

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

URČENÍ POLOHY STANIC V SÍTI INTERNET POMOCÍ
PŘENOSOVÉHO ZPOŽDĚNÍ

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. MICHAEL HORÁK

BRNO 2013



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ**
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

URČENÍ POLOHY STANIC V SÍTI INTERNET POMOCÍ PŘENOSOVÉHO ZPOŽDĚNÍ

GEOLOCATION OF INTERNET NODES BASED ON COMMUNICATION LATENCY

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. MICHAEL HORÁK

VEDOUCÍ PRÁCE
SUPERVISOR

doc. Ing. DAN KOMOSNÝ, Ph.D.

BRNO 2013



**VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ**

**Fakulta elektrotechniky
a komunikačních technologií**

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student: Bc. Michael Horák

ID: 119436

Ročník: 2

Akademický rok: 2012/2013

NÁZEV TÉMATU:

Určení polohy stanic v síti Internet pomocí přenosového zpoždění

POKYNY PRO VYPRACOVÁNÍ:

Nastudujte a analyzujte dílčí faktory způsobující zpoždění při přenosu dat mezi stanicemi v IP sítích. Dále se seznamte s experimentální sítí PlanetLab – <http://www.planet-lab.org/>. Naprogramujte lokační algoritmus CBG (Constraint-Based Geolocalization). K určení pozice použijte sférické souřadnice. Pomocí stanic sítě PlanetLab ověřte správnou činnost realizovaného algoritmu.

DOPORUČENÁ LITERATURA:

- [1] GUEYE, B. ZIVIANI, A., CROVELLA, M., FDIDA, S. Constraint-Based Geolocation of Internet Hosts. ACM Internet Measurement Conference 2004. ACM SIGCOMM, 2004.
- [2] PERCACCI, R., VESPIGNANI, A. Scale-free behavior of the Internet global performance. The European Physical Journal B – Condensed Matter and Complex Systems. Springer-Verlag, 2003.
- [3] PlanetLab Consortium. PlanetLab: An open platform for developing, deploying, and accessing planetary-scale services. URL: <<http://www.planet-lab.org>> [cit. 10. 10. 2011].

Termín zadání: 11.2.2013

Termín odevzdání: 29.5.2013

Vedoucí práce: doc. Ing. Dan Komosný, Ph.D.

Konzultanti diplomové práce:

prof. Ing. Kamil Vrba, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Tato práce se zabývá zjištění geografické lokace stanice v síti internet pomocí měření zpoždění a implementací metody Constraint-Based Geolocation. Postupně je rozebrána problematika zpoždění v počítačových sítích a jeho měření. Stručně jsou popsány různé možnosti geolokace cílových stanic, přičemž je kladen důraz především na popis metody CBG. Dále je vysvětlen převod sférických souřadnic do dvojrozměrného prostoru, čehož je využito při implementaci geolokační metody. V praktické části je při popisu implementovaného programu v jazyce JAVA navazováno na problematiku popsanou v předchozích kapitolách. V rámci práce byly implementovány a porovnány dvě geolokační metody. Ze získaných výsledků je dále navržena hybridní geolokační metoda, jenž tyto dvě předchozí kombinuje. V zhodnocení jsou uvedeny výsledky implementace a jejich porovnání s prameny, ze kterých tato práce vycházela.

KLÍČOVÁ SLOVA

Geolokace, CBG, Constraint-Based Geolocation, zpoždění, latence, RTT, IP, mapové projekce

ABSTRACT

This thesis covers the topic of determination of geographical location of a host in internet network while utilizing measurement of the end to end delay and implementation of Constraint-Based Geolocation. Gradually I go through issue of the delay in computer networks and ways how to measure it. Next chapter describes a few ways to geolocate host in internet network with emphasis on the CBG method. Another chapter is dedicated to describing a way to project spherical coordinates to the two dimensional space, which has been used in implementation of geolocation method. Chapter about implementation builds upon the facts given in previous chapters while functions of the program written in the JAVA programming language are being explained. Two similar geolocation methods were implemented. By comparing the results gained by implementation, new method of geolocation is proposed and devised. It combines properties of both previous methods. There are results of the implemented methods and their comparison to the one of the source documents used in creation of this thesis in summary section.

KEYWORDS

Geolocation, CBG, Constraint-Based Geolocation, delay, latency, RTT, IP, map projections

HORÁK, Michael *Určení polohy stanic v síti Internet pomocí přenosového zpoždění*: diplomová práce. BRNO: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2013. 70 s. Vedoucí práce byl doc. Ing. Dan Komosný, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Určení polohy stanic v síti Internet pomocí přenosového zpoždění“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

BRNO

.....
(podpis autora)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu doc. Ing. Danu Komosnému, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

BRNO

.....

(podpis autora)

OBSAH

Úvod	11
1 Zpoždění v sítích IP	12
1.1 Deterministické a stochastické zpoždění	12
1.2 Zpoždění na koncových uzlech	12
1.3 Zpoždění na mezilehlých uzlech	12
1.4 Zpoždění na transportních linkách	13
1.5 Měření zpoždění	14
1.5.1 Jednocestné zpoždění	14
1.5.2 Obousměrné zpoždění	15
1.6 Protokol ICMP	15
1.6.1 PING	16
1.6.2 Traceroute a tracert	18
2 Geolokace	20
2.1 Využití geolokace	20
2.2 Dělení IP geolokačních metod	20
2.2.1 Pasivní metody	21
2.2.2 Aktivní metody	21
2.3 Metoda Constraint-Based Geolocation	22
2.3.1 Multilaterace v CBG	22
2.3.2 Převod zpoždění na geografickou vzdálenost	23
2.3.3 Zjištění polohy cílové stanice	24
2.3.4 Důsledky podhodnocení a nadhodnocení vzdálenosti	25
3 Platforma PlanetLab	26
3.1 Struktura PlanetLab	27
3.2 Správa serverů	28
3.3 Využití PlanetLabu pro geolokaci	28
4 Mapové projekce	29
4.1 Válcové projekce	29
4.2 Ekvidistantní válcová projekce	30
4.3 Vliv EVP na tvar oblasti	31
5 Návrh řešení a implementace metody CBG	34
5.1 Měření dat pro Geolokaci	34
5.1.1 Vybrané uzly Planetlabu	34

5.1.2	Bash skripty a třídění dat	35
5.1.3	Soubory s naměřenými daty	36
5.2	Algoritmus pro zjištění nejvhodnější přímky	37
5.2.1	Matematika na pozadí	37
5.2.2	Implementace v programu	38
5.3	Hlavní program	40
5.3.1	Třídy	40
5.3.2	Metoda <i>main</i> a průběh výpočtu cílových souřadnic	45
5.4	Porovnání výsledků metod SOI a CBG	49
5.4.1	Hybrid metod SOI a CBG	49
6	Diskuse výsledků	52
7	Závěr	55
	Literatura	56
	Seznam symbolů, veličin a zkratk	58
	Seznam příloh	59
A	Obsah přiloženého média	60
B	Seznam uzlů sítě Planetlab	61
C	Seznam stanic pro účely geolokace	64
D	Vývojové diagramy	67
E	Chyba určené polohy pro geolokační metody	69

SEZNAM OBRÁZKŮ

1.1	Modely ISO/OSI a TCP/IP	13
1.2	Ukázkový výstup programu PING v OS Linux distribuci openSUSE	
12.2	17
1.3	Ukázkový výstup programu traceroute v OS Linux na geolokačním serveru VUT Bětko	19
2.1	Multilaterace v CBG s omezením geografické vzdálenosti	23
2.2	Příklad grafu závislosti zpoždění na geografické vzdálenosti pro jednu sondu	24
2.3	Důsledek nadhodnocených a podhodnocených geografických vzdáleností. (a) Nadhodnocené vzdálenosti. (b) Podhodnocené vzdálenosti. (c) Kombinace předchozích.	25
3.1	Lokality sítě PlanetLab na mapě světa, obrázek čerpán ze zdroje[18]	26
3.2	Evropské lokality sítě PlanetLab, obrázek čerpán ze zdroje[19]	27
3.3	Část seznamu lokalit, čerpáno z[18]	28
4.1	Slepá mapa světa vytvořená EVP	30
4.2	Závislost velikosti zkrácení na zvětšující se vzdálenosti od rovníku, čerpáno z [11]	31
5.1	Uzly PlanetLab použité pro získání dat	34
5.2	Zjednodušený vývojový diagram skriptu Bash pro získání dat	36
5.3	Obsah souboru s naměřenými daty	37
5.4	Obsah souboru <i>seznam_stredu_complete.txt</i>	37
5.5	Obsah souboru <i>Originalni_pozice.txt</i>	37
5.6	Body B0–4 zobrazené v kartézské soustavě souřadnic	38
5.7	Vývojový diagram algoritmu výběru nejlepší přímkou	40
5.8	Graficky zpracovaný výstup algoritmu programu pro jednu stanici (IP: 194.47.148.170, $a \approx 0,026$, $b \approx 2.915$)	41
5.9	UML diagram tříd hlavního programu	42
5.10	Vývojový diagram metody <i>nastavZpozdeniCilumOstry</i>	44
5.11	Vývojový diagram střední části programu, práce s oblastmi	47
5.12	Záhlaví a první čtyři řádky výstupu programu	47
5.13	Lokalizace čtyř stanic	48
5.14	Kumulativní distribuční funkce. a) S omezením vzdálenosti b) Bez omezení vzdálenosti	50
5.15	Závislost mediánu chyby metody na prahové konstantě	51
6.1	Porovnání implementovaných geolokačních metod s [14]	52
6.2	Nevhodně zvolená stanice, která nemohla být díky poloze dostatečně přesně lokalizována	53

D.1	Vývojový diagram úvodní části programu, příprava polí	67
D.2	Vývojový diagram koncové části programu, počítání chyby metody a výpis výsledků	68

SEZNAM TABULEK

1.1	Význam parametrů programu ping v jednotlivých OS	17
1.2	Význam vybraných parametrů programů tracert/traceroute v jednotlivých OS	19
4.1	Zkreslení EVP na různých zeměpisných šířkách	32
4.2	Poloha bodů udaná sférickými a ortogonálními souřadnicemi a jejich dosazení do rovnice elipsy.	33
5.1	Statistické údaje z chyb určených poloh	49
6.1	Statistické údaje z chyb určených poloh pro údaje s chybou menší než 600 km	54
B.1	Seznam pužitých uzlů sítě Planetlab	63
C.1	Seznam stanic pro účely geolokace	66
E.1	Chyba určení výsledných souřadnic	70

ÚVOD

Trendem moderní doby jsou tzv. location awerness aplikace, což jde do češtiny hrubě přeložit jako aplikace, které znají geografickou polohu uživatele a dokážou na to reagovat. Využít známé polohy lze různými způsoby – zobrazení počasí pouze pro daný region, nabídnutí relevantních dopravních informací, omezení obsahu či nejčastěji cílení reklamy.

Zařízení připojené do sítě internet lze lokalizovat několika způsoby. Existují metody pracující se znalostí síťové topologie, metody pracující s databázemi (pasivní) a metody měřící vlastnosti sítě, převážně obousměrné zpoždění (aktivní).

Tato práce se věnuje lokaci stanic v síti internet pomocí měření zpoždění. Toho docílí využitím aktivní geolokační metody Constraint-Based Geolocation.

V kap. 1 je rozebrána problematika zpoždění v IP sítích. Kde vzniká, jeho charakter, jaké faktory mají vliv na jeho velikost a jak ho lze měřit. Kap. 2 rozebírá aktivní i pasivní internetové geolokační metody. Taky je zde popsán princip funkce metody Constraint-Based Geolocation.

Kap. 3 pojednává o síti PlanetLab, jenž bude využita pro budoucí měření hodnot obousměrného zpoždění. Je zde popsán hlavně charakter, struktura sítě a její využití pro geolokační účely. V kap. 4 je popsána metoda mapového zobrazení, jenž je v této práci použito pro převod Země jako trojrozměrného tělesa do dvojrozměrné plochy.

Samotné implementaci geolokační metody Constraint-Based Geolocation je věnována kap. 5. Nachází se zde návrh řešení, popis postupu měření dat a také popis samotného programu a jeho součástí. Vychází se zde z poznatků uvedených v předchozích kapitolách. Vzhledem k minimálním rozdílům ve funkcionalitě, jsou zde porovnány dvě aktivní geolokační metody (Constraint-Based Geolocation a Speed of Internet) pro dvě různě velké množiny sond. Výsledky onoho porovnání vedly k návrhu vlastní geolokační metody kombinující vlastnosti obou výše zmíněných, viz kap. 5.4.

Výsledky implementace jsou uvedeny v kap. 6. Přesnost implementovaných metod je rozebrána již dříve, zde je porovnávána výsledná hybridní metoda s pramenem [14], který byl jedním z výchozích pro tuto práci. Dále je zde zdůrazněno několik požadavků na správné, resp. s dostatečnou přesností, určení polohy lokalizované stanice.

1 ZPOŽDĚNÍ V SÍTÍCH IP

Jedná se o dobu, mezi odesláním a přijetím zprávy koncovými uzly. Někdy se pro něj používá výrazu latence. V této kapitole bude stručně vysvětleno, kde a proč vzniká.

Na celkové velikosti zpoždění se podílí řada faktorů. Ty lze kategorizovat dle toho, kde působí nebo jejich charakteru. Dle oblasti působení se dělí na zpoždění na prvcích koncových a mezilehlých a na zpoždění na přenosových liniích[2]. Ve všech částech informačního řetězce pak lze zpoždění dělit na deterministické a stochastické.

1.1 Deterministické a stochastické zpoždění

Při opakování měření jsou zpravidla naměřené hodnoty zpoždění podobné. Hodnotu, pod kterou zpoždění nikdy neklesne lze nazvat deterministickým zpožděním a je možné ji odhadnout z teoretických poznatků. Vše nad tuto hodnotu je tzv. stochastickým zpožděním. To je závislé na zatížení sítě a jiných náhodných jevech[4].

1.2 Zpoždění na koncových uzlech

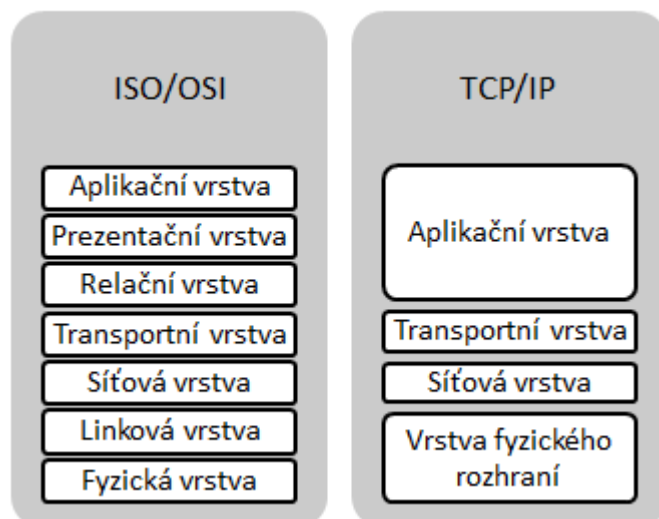
Koncovým uzlem se rozumí zdroj nebo příjemce zprávy. Odesílaná data, ze kterých se zpráva skládá, projdou postupně všemi vrstvami referenčního modelu, směrem odshora dolů, kde se opatří adresami a záhlavími příslušícím daným vrstvám[6]. Pro počítačové sítě je jím zpravidla model TCP/IP, pro vysvětlení je ovšem někdy vhodnější použít obecný ISO/OSI.

Příjemce zprávy provede tytéž úkoly co odesílatel, jen v obráceném pořadí. Postup referenčního modelu je vidět na obr. 1.1.

Zpoždění zde vzniká přípravou dat k odeslání, případně čekáním na přístup k přenosovému médiumu.

1.3 Zpoždění na mezilehlých uzlech

Mezilehlým uzlem se obvykle rozumí aktivní síťové prvky, které jsou použity k propojení odesílatele a příjemce dat. Pracují na různých vrstvách modelu ISO/OSI. Na fyzické vrstvě opakovací a rozbočovací (repeatery a HUBy), na linkové přepínače (switche), na síťové směrovače (routery) a i některé přepínače[2, 7]. Pokud je ve spoji zařazen firewall, pak může pracovat i na vyšších vrstvách.



Obr. 1.1: Modely ISO/OSI a TCP/IP

Z principu funkce je patrné, že na čím vyšší vrstvě prvek funguje, tím musí provádět více úkonů (odstraňování, vyhodnocování a následné opětovné přidávání hlaviček), a tím se zvyšuje čas, který stráví datová jednotka na cestě k cíli.

Zpoždění způsobené prvky na fyzické vrstvě je zanedbatelné. Zpoždění vnesené přepínači je zpravidla v jednotkách μs , pro přepínače je to hodnota v rozmezí 10–100 μs [2, 17].

Dalším faktorem, který je potřeba zohlednit, je zatížení síťových prvků. Při větším zatížení sítě musí někdy datové jednotky čekat, ve vstupních nebo výstupních frontách prvku, na uvolnění prostředků. Nastane-li tato situace, pak se zpoždění vnesené prvkem může zvýšit až na několikanásobek nezatížené hodnoty.

1.4 Zpoždění na transportních linkách

Transportní linky ovlivňují výsledné zpoždění rychlostí propagace signálu v médiu a svojí délkou.

Délka linky spojující koncové uzly je zpravidla větší, než délka vzdušné čáry spojující tyto body. To je zapříčiněno tím, že kabelové trasy zpravidla kopírují jiné stavby, jako jsou například dálnice, sloupy vysokého napětí a železniční dráhy ¹.

Na celkovou vzdálenost má následně i vliv směrovacích politik. Zprávy se nemusí šířit nejkratší možnou cestou, například z ekonomických důvodů či z důsledku rozložení zátěže[17].

¹Na území České republiky mimo jiné působí společnost ČD telematika. Mapa jejich infrastruktury je k dispozici na <http://www.cdt.cz/cz/infrastruktura-97/>

V optických kabelech, ze kterých se převážně skládají páteřní sítě se signál šíří téměř 2/3 rychlosti světla ($c = 299\,792\,458$ m/s)[2, 17]. To dává v přepočtu rychlost šíření signálu v optickém médiu 100 km/ms. V metalickém médiu se signál šíří 3/4 rychlosti světla.

Známe-li kromě rychlosti šíření signálu ještě jeden údaj, délku vedení nebo dobu šíření signálu, pak jsme úpravou vztahu $s = v \cdot t$ schopni dopočítat údaj zbývajících. V onom vztahu s značí délku vedení, v rychlost šíření signálu a t dobu šíření signálu.

Vzhledem k tomu, že jsme schopni spočítat zpoždění na přenosové lince z teoretického vztahu, považujeme toto za deterministické.

1.5 Měření zpoždění

Níže jsou rozebrány některé metody, jak lze měřit latence v IP sítích a prostředky, které využívají. Jedná se o měření jednocestného a obousměrného zpoždění pomocí programů využívajících ICMP protokol, PING a traceroute.

1.5.1 Jednocestné zpoždění

Anglicky one-way delay. Pojem jednocestné zpoždění představuje časový rozdíl od vzniku a vyslání zprávy ve zdrojové stanici po přijetí a zpracování ve stanici cílové. Aby to bylo možné, zpráva musí obsahovat časové razítko odeslání. Cílová stanice pak jen spočítá rozdíl od aktuálního času. K tomu potřebují být obě stanice co nejpřesněji časově synchronizované.

Nejčastěji se pro synchronizaci používá network time protocol (NTP). Stanice svůj čas synchronizuje s NTP serverem/servery, ty jsou hierarchicky děleny dle přesnosti. Používá se notace stratum N , kde N značí úroveň serveru, nejpřesnější servery mají $N=1$, servery přebírající svůj čas od nich mají $N=2$ atd. Jak již je napsáno výše, síťová komunikace probíhá s určitým zpožděním a to má vliv i na synchronizaci NTP[9]. Dle polohy stanic lze uvažovat hodnoty:

- 0,5–2 ms pro LAN
- 10–100 ms pro WAN

Pro většinu aplikací daná přesnost bohatě dostačuje, pro lokalizaci stanic v síti IP pomocí měření zpoždění je však nevyhovující. Vyjdeme-li z poznatků udaných v kap.1.4, každá milisekunda rozdílu v synchronizaci znamená v důsledku možné zvětšení vzdálenosti stanic o 100 km.

Globální družicový polohový systém (Global Positioning System - GPS) dává možnost, jak spolehlivě synchronizovat stanice. Naneštěstí jeho použití je limitováno tím, že přijímač potřebuje mít neustále spojení s několika satelity, čehož se v budovách zpravidla těžko dosahuje.

Výše zmíněné faktory mají vliv na to, že v praxi se jednocestné zpoždění měří velmi obtížně a je potřeba sáhnout po jiných prostředcích.

1.5.2 Obousměrné zpoždění

Anglicky round-trip (delay) time (RTT). Toto zpoždění je vlastně součet jednocestných zpoždění ze zdroje k cíli a zpět. Jeho měření má oproti měření jednocestného zpoždění určité výhody ale také existuje několik faktorů, které je třeba při odečítání výsledné hodnoty vzít v potaz.

Za velkou výhodou je považována jednoduchost metody. K jejímu provedení stačí mít přístup pouze k jedné stanici, na rozdíl od měření jednocestného zpoždění, kde je potřeba mít zajištěnou obsluhu i na cílové stanici, což zpravidla nebývá možné. Díky tomuto faktu zde odpadá problém i s časovou synchronizací, veškerá práce s časem probíhá právě na jednom místě.

Výhodou ale i kamenem úrazu může být fakt, že měření obousměrného zpoždění převážně spoléhá na protokol ICMP (Internet Control Message Protocol). ICMP je součástí balíku protokolů internetu, proto ho stanice v této síti podporují. Jeho funkci ale lze, například z bezpečnostních důvodů omezit, takže ne vždy se na jeho funkčnost lze spolehnout. O protokolu a programech ho využívajících je více napsáno v kap.1.6. Využití ICMP ovšem není jedinou možností měření RTT. Některé proprietární aplikace používají systém založený na periodickém odesílání sondovacích dat pomocí protokolu UDP.

Jak již bylo napsáno výše, RTT je čas, který uplyne od odeslání zprávy k jejímu přijetí. Při směrování paketů mohou nastat dvě situace, které ovlivní výslednou hodnotu RTT. První z nich je load balancing, česky rozložení zátěže. Pakety v jednom směru používají dvě a více cest. Dále může RTT ovlivnit asymetrické směrování (asymmetric routing). Dojde-li k němu, pak pakety od zdrojové stanice k cílové putují jinou cestou, než naopak. V obou případech se trasa mění a s tím i zpoždění - pakety nejsou směrovány nejkratší cestou.

1.6 Protokol ICMP

Jak již bylo napsáno výše, ICMP je jedním ze stěžejních protokolů internetu. Pracuje na síťové vrstvě, stejně jako protokol IP. Je využíván operačními systémy počítačů propojených do sítě k odesílání chybových zpráv, například o nedostupnosti služby, odpovídá na IP operace atp. Dále se používá k různým dotazům.

ICMP bývá často na síťových prvcích blokován. Může tomu tak být z bezpečnostních důvodů – pomocí programu traceroute lze skenovat propojení vnitřní sítě.

Nejčastěji je tento jev vidět ve velkých tranzitních sítích. Dalším důvodem může být špatné (ale i záměrné)² nastavení firewallu.

Níže je stručný výpis nejpoužívanějších ICMP zpráv.

- **Echo request** – Dotaz na prvek pracující na síťové vrstvě. Každý prvek na této vrstvě (má přidělenou IP adresu) by měl na tento dotaz reagovat.
- **Echo reply** – Odpověď na echo request.
- *** unreachable** – Odpověď obsahující nedostupnost *, kde * může být např.:
 - **Net** – Cílová síť, např. když výchozí brána nezná cestu do cílové sítě.
 - **Host** – Cílová stanice .
- **Redirect** – Odpověď oznamující přesměrování.
- **Time Exceeded** – Vypršel časový limit. Zpráva *Time to Live exceeded in Transit* znamená, že paket nebyl doručen z důvodu snížení pole TTL na 0³.

1.6.1 PING

PING (angl. Packet InterNet Groper) je program, jenž nachází nejvíce uplatnění v testování dostupnosti stanic v IP sítích a měření RTT. Další možností je využít ping jako sondu, pro zjištění nastavení velikosti IP paketu na trase.

Princip funkce je následující. Periodicky odesílá ICMP pakety *Echo request* na cílovou stanici. Ta odešle odpověď *Echo response*. Pro všechny pakety se měří čas uplynulý od odeslání k přijetí. Výsledkem je RTT. Dále se zaznamenává případná ztráta/nedoručení paketu. Výstupem programu je přehled, ve kterém je procento ztracených paketů, minimální, střední a maximální RTT a někdy i směrodatná odchylka od střední hodnoty.

Ping je součástí většiny operačních systémů. Syntaxe příkazu pro MS Windows⁴ a Linux s vybranými parametry je k vidění níže⁵. Význam vybraných⁶ parametrů je vysvětlen v tab. 1.1.

MS Windows: `ping [-t] [-a] [-n] [-f] [-i] TargetName`

²Příkladem může být stavový firewall. Tento typ firewallu propustí do sítě pouze pakety náležící už sestavenému spojení uvnitř sítě. To komplikuje získání odpovědi od stanice za tímto firewallem.

³Pole TTL v hlavičce paketu obsahuje celé číslo, které je po průchodu síťovým prvkem dekrementováno o jedna. Když dosáhne hodnoty 0, paket se zahodí a vygeneruje se příslušná zpráva. Jedná se o mechanismus, který může síťovým prvkům odlehčit při vytvoření směrovací smyčky.

⁴Standartní PING v MS Windows měří RTT s přesností na jednotky milisekund. Pro přesnější měření je použit pokročilejších programů, jako je např. trueping

⁵Další možností je využít on-line nástrojů, např. on-line pingu dostupného z: <<http://ping.eu/>>

⁶Kompletní seznam je možné zjistit v MS Windows zadáním příkazu „ping“. Pro Linuxové OS je krátký výpis k vidění po zadání „ping -help“, kompletní manuál se zobrazí po zadání „man ping“.

Linux: ping [-n] [-v] [-i] [-t] host [packetsize] [-c]

MS Windows	Linux	Význam parametru
-n	-c	Počet, kolikrát se ping provede.
-t		Ping se provádí, dokud není ukončen klávesou zkratkou Ctrl+C.
-a	-n	Překládá IP adresy na názvy hostitelů.
-f		Zakáže fragmentaci paketu.
-i	-t	Nastaví hodnotu pole TTL na zadanou hodnotu.
	-v	Verbose režim, vypisuje i jiné než jen ICMP <i>Echo response</i> pakety.
	packetsize	Nastaví velikost odchozího paketu.
TargetName	host	Název nebo IP adresa, na kterou má být proveden ping.

Tab. 1.1: Význam parametrů programu ping v jednotlivých OS

Vzhledem k tomu, že PING ke své funkci využívá primárně protokolu ICMP, jsou jeho omezení totožná.

Na obr.1.2 je vidět výstup programu PING. Cílový server byl wikipedia.org, PING se opakoval šestkrát (viz sloupec Linux, parametr -c v tab. 1.1). Všech šest paketů bylo úspěšně přijato zpět, což je ve výpisu patrné z: „0% packet loss“. Dále je vidět, že serveru je přiřazena IP adresa 208.80.152.201, jak dlouho se program prováděl a hodnoty RTT v pořadí minimální, střední, maximální a směrodatná odchylka od střední hodnoty.

```
Ork@linux-gsja:~> ping wikipedia.org -c 6
PING wikipedia.org (208.80.152.201) 56(84) bytes of data:
64 bytes from wikipedia.org (208.80.152.201): icmp_seq=1 ttl=47 time=156 ms
64 bytes from wikipedia.org (208.80.152.201): icmp_seq=2 ttl=47 time=158 ms
64 bytes from wikipedia.org (208.80.152.201): icmp_seq=3 ttl=47 time=155 ms
64 bytes from wikipedia.org (208.80.152.201): icmp_seq=4 ttl=47 time=206 ms
64 bytes from wikipedia.org (208.80.152.201): icmp_seq=5 ttl=46 time=152 ms
64 bytes from wikipedia.org (208.80.152.201): icmp_seq=6 ttl=46 time=153 ms

--- wikipedia.org ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5008ms
rtt min/avg/max/mdev = 152.432/163.808/206.157/19.061 ms
```

Obr. 1.2: Ukázkový výstup programu PING v OS Linux distribuci openSUSE 12.2

1.6.2 Traceroute a tracert

Dalším z programů, které jsou často používaným nástrojem při diagnostice IP sítí je traceroute. Traceroute je standartní součástí OS Linux, v MS Windows je obsažen funkčně stejný program tracert. Užívání programů traceroute/tracert se v některých zdrojích říká trasování.

Stejně jako program PING využívá ke své funkci protokol ICMP – hlavně zpráv *Echo request* a *Time to Live exceeded in Transit*. Druhé jmenované jsou vygenerovány vždy, když je zařízením pracujícím na třetí/druhé (síťové, ISO-OSI/TCP-IP) zahozen paket z důvodu snížení obsahu pole TTL v hlavičce na 0. Program traceroute vyšle několik paketů, tento počet je možné nastavit, implicitně se však posílají tři pakety. První skupina paketů má TTL implicitně nastaveno na 1. Na prvním prvku pracujícím na síťové vrstvě (zvláště v malých sítích to je výchozí brána) jsou zahozeny a stanici inicializující spojení stanici je odeslána ICMP zpráva *Time to Live exceeded in Transit*. Postně se TTL inkrementuje. Tím se pakety dostávají dále sítí, až dorazí k cílové stanici.

Syntaxe pro příkazovou řádku jednotlivých OS s vybranými parametry⁷ je k vidění níže. Význam těchto parametrů je vysvětlen v tab. 1.2.

MS Windows: `tracert [-d] [-h] [-S] [-4] [-6] TargetName`

Linux: `traceroute [-46ITU] [-f] [-m] [-s] [-q] host`

Výstupem programu traceroute/tracert je výpis po řádcích. Na každém řádku je jméno síťového uzlu, jeho IP adresa a q hodnot RTT, význam hodnoty q je vysvětlen v tab. 1.2. Každý nový řádek znamená zvýšení hodnoty TTL o jedna a jedná se tedy o síťový prvek vzdálený o jeden skok dál, než na předchozím řádku.

Na obr. 1.3 je vidět ukázkový výstup programu traceroute. Příkaz byl proveden v OS Linux na jednom ze serverů VUT. Cílová adresa byla zvolena wikipedia.org, argument `-f 3` znamená, že první paket byl odeslán s hodnotou TTL rovnou třem. Z toho vyplývá, že první síťový prvek, který odešle zprávu *Time to Live exceeded in Transit* bude až třetí v pořadí. K dosažení cíle bylo potřeba projít patnácti směrovacími prvky.

⁷ Kompletní seznam je možné zjistit v MS Windows zadáním příkazu „tracert“. Pro Linuxové OS je krátký výpis k vidění po zadání „traceroute“ nebo pro některé distribuce „/usr/sbin/traceroute“, kompletní manuál se zobrazí po zadání „man traceroute“, případně „man /usr/sbin/traceroute“.

MS Windows	Linux	Význam parametru
-d	-n	IP adresám nebudou přiřazovány názvy hostitelů.
-h	-m	Maximální počet směřování při hledání cíle, tedy max. hodnota TTL. Pro oba OS je implicitně nastaveno 30.
-S	-s	Zdrojová adresa použitá při trasování.
-4/6	-4/6	Nastaví příslušnou verzi protokolu IP.
	-I/T/U	Nastaví protokol, který se použije při trasování. Volby jsou I-ICMP, T-TCP a U-UDP.
	-f	Nastaví první hodnotu TTL, která se následně inkrementuje, přednastaveno pro oba OS je 1.
	-q	Nastaví počet odeslaných paketů na jeden dotaz, přednastaveno pro oba OS je 3.
TargetName	host	Název nebo IP adresa, na kterou má být proveden traceroute.

Tab. 1.2: Význam vybraných parametrů programů tracert/traceroute v jednotlivých OS

```
[xhorak07@localization ~]$ traceroute wikipedia.org -f 3
traceroute to wikipedia.org (208.80.152.201), 30 hops max, 60 byte packets
 3  hp-tech.net.vutbr.cz (147.229.254.137)  2.669 ms  2.657 ms  2.645 ms
 4  pe-ant.net.vutbr.cz (147.229.252.201)  1.939 ms  2.118 ms  2.283 ms
 5  gw-ant6.net.vutbr.cz (147.229.253.233)  3.040 ms  3.470 ms  3.885 ms
 6  r98-bm.cesnet.cz (147.229.252.17)  0.792 ms  0.897 ms  1.016 ms
 7  195.113.179.150 (195.113.179.150)  3.148 ms  3.347 ms  3.440 ms
 8  vix2.swip.net (193.203.0.105)  6.686 ms  6.541 ms  6.546 ms
 9  wen3-core-1.tengigabiteth4-1-0.tele2.net (130.244.71.76)  7.230 ms  6.810 ms  6.923 ms
10  cbv-core-3.pos0-12-1-0.tele2.net (130.244.207.53)  28.867 ms  27.811 ms  27.799 ms
11  ash1-core-2.pos2-0-0.tele2.net (130.244.207.206)  106.833 ms  107.397 ms  107.386 ms
12  ash1-peer-1.xe-0-2-0-unit0.tele2.net (130.244.64.147)  113.981 ms  114.077 ms  114.067 ms
13  130.244.6.243 (130.244.6.243)  102.977 ms  102.916 ms  102.908 ms
14  ae0.cr2-eqiad.wikimedia.org (208.80.154.194)  101.093 ms  101.181 ms  101.156 ms
15  xe-1-1-0.crl-sdtpa.wikimedia.org (208.80.154.214)  127.507 ms  127.556 ms  127.524 ms
16  wikipedia-lb.pmtpa.wikimedia.org (208.80.152.201)  130.840 ms  130.895 ms  130.841 ms
```

Obr. 1.3: Ukázkový výstup programu traceroute v OS Linux na geolokačním serveru VUT Bětká

2 GEOLOKACE

Geolokací se rozumí metoda určování polohy na zemském povrchu. Jejím výstupem mohou být například adresa (stát, město, ulice...), lokalita (hory, řeka, park...) nebo sférické souřadnice. K tomu lze využít různé metody, GPS přijímače, základnové stanice mobilních operátorů, znalost polohy wi-fi sítí atp. Na zvolených prostředcích závisí přesnost. I civilní GPS je schopna dosáhnout přesnosti jednotek metrů¹, pokud má dobrý příjem. Při určování polohy podle SSID wi-fi sítí lze dosáhnout přesnosti na nízké desítky metrů, podobně je tomu při využití základnových stanic, u nich výsledná přesnost závisí na kvalitě infrastruktury v oblasti.

Tato práce se věnuje geolokaci v síti pracujících s protokolem IP. Tyto metody se souhrnně označují jako IP geolokace. V této kapitole budou popsány dva základní typy IP geolokací s jejich nejznámějšími metodami. Blíže se budu věnovat metodě Constraint-Based Geolocation, zkráceně CBG, jejíž implementace je cílem této práce.

2.1 Využití geolokace

Od moderních internetových služeb a aplikací se žádá takzvaná location awareness neboli znalost polohy. V praxi to znamená, že svou činnost uzpůsobí dle lokality, kde se dané koncové zařízení nachází. Může se jednat například o nastavení jazyka nebo omezení obsahu. Typickým příkladem druhého zmíněného, je na videoseveru youtube.com „Toto video není dostupné ve vaší zemi“. Mimo IP geolokační metody popsány výše a v kap. 2.2 mohou aplikace zjistit údaje z nastavení koncového zařízení – jazyk a čas ve webovém prohlížeči nebo třeba nastavení uživatelského účtu (e-mail, sociální síť...).

Hlavním důsledkem pro koncového uživatele je, s možným využitím jiných zdrojů dat, typicky historie chování či cookies², cílená reklama. Kromě ní je možné například zjistit jízdní řády z okolních zastávek MHD, zjistit kulturní akce v okolí, nechat si doporučit restauraci v cizím městě či jen načíst počasí.

2.2 Dělení IP geolokačních metod

Jak název napovídá, IP geolokace se užívá v sítích postavených na protokolu IP, respektive v jejím největším a nejznámějším zástupci, internetu.

¹ Čehož civilní sektor využil například při geolokační hře geocaching. Viz <<http://www.geocaching.cz/news.php>> a <<https://www.geocaching.com/>>

² Cookie je malý soubor uložený u uživatele, který je používán jako úložiště osobních dat při HTTP komunikaci klient-server.

Existují dva způsoby, jak lze při IP geolokaci postupovat. První vychází z faktu, že IP adresy či doménová jména jsou pevně spjatá s určitou organizací a zpravidla uloženy v relačních databázích. Metody pracující s těmito daty jsou souhrnně označovány jako pasivní. Opakem pasivních metod jsou metody aktivní, které využívají k zjištění polohy koncové stanice měření vlastností IP sítě[15]. Nejčastěji měřenou veličinou je přenosové zpoždění.

2.2.1 Pasivní metody

Tyto metody spoléhají na manuálně udržované databáze, jejichž klíči jsou převážně IP adresy a doménová jména. Dle nich se určí výsledná poloha. Využívá se toho, že IP adresy jsou po blocích pevně přiděleny subjektům organizací IANA. Doménová jména jsou pak zaregistrována u místních registrátorů. V obou případech lze poměrně jednoduše vytvořit záznam obsahující IP adresu nebo jejich rozsah, doménu a organizaci obě předchozí využívající.

Databáze jsou dvojího druhu, komerční a komunitní[15]. Nevýhoda komerčních databází tkví v tom, že za jejich využívání se platí. Nabízí sice své služby i zdarma, ty jsou ale zpravidla omezené. Jelikož je udržování aktuálních databází placenými pracovníky velmi nákladné, vznikají v duchu webu 2.0 komunitní databáze. Jejich využití může být levnější, nikdo ale nezaručuje kvalitu obsahu.

S oběma druhy databází je pak spjat problém s aktualitou. Výše je napsáno, že udržování databází je nákladný proces. Je to zapříčiněno hlavně přerozdělováním IP adres z důvodu efektivity.

Nejnámější databáze jsou MaxMind GeoIP a WHOIS.

2.2.2 Aktivní metody

Jak již název napovídá, tyto metody přistupují k určení polohy cílové stanice aktivně, převážně měřením určitých vlastností sítě. Všechny níže zmíněné metody využívají měření zpoždění.

Pro další vysvětlování je nejprve nutné definovat několik pojmů:

- **Cíl <C>** – Označení stanice, jejíž polohu chceme pomocí geolokace zjistit. Angl. *target*.
- **Referenční bod <R>** – Označení pro stanici se známou geografickou polohou, k níž vztahujeme polohu cílů. Angl. *landmark*.
- **Sonda <S>** – Označení pro stanici se známou geografickou polohou, z níž jsou prováděna aktivní měření. Angl. *probe*.

I když je tento přístup levnější a aktuálnější, je potřeba mít na paměti omezení těchto metod. Pro dostatečnou přesnost u některých metod potřebujeme mít dosta-

tečný počet sond nebo referenčních bodů. Jsou-li tyto metody prostředkem k tvoření nějaké databáze stanic, pak jsou vhodné je pro nepřemísťované stanice, jako jsou například servery. Stanice připojené k IP síti přes satelitní spoj, mobilní připojení, wi-fi či jinou mobilní technologii mohou a mění svoji polohu.

Mezi aktivní metody se řadí shortest PING, geo PING, speed of internet (SOI), octant a geoweighth.

2.3 Metoda Constraint-Based Geolocation

V této kapitole je obecně popsána Constraint-Based Geolocation, dále jen CBG. Bude vysvětleno, jak je CBG schopné přepočítat naměřená zpoždění na geografické vzdálenosti, ty využít k multilateraci a tím určit polohu cílové stanice v internetu. Dále je vysvětleno použití omezení vzdálenosti (distance constraints), klíčové vlastnosti CBG, pro přesnou transformaci naměřeného zpoždění na geografickou vzdálenost. Pokud není napsáno jinak, v této kapitole jsem čerpal ze [14].

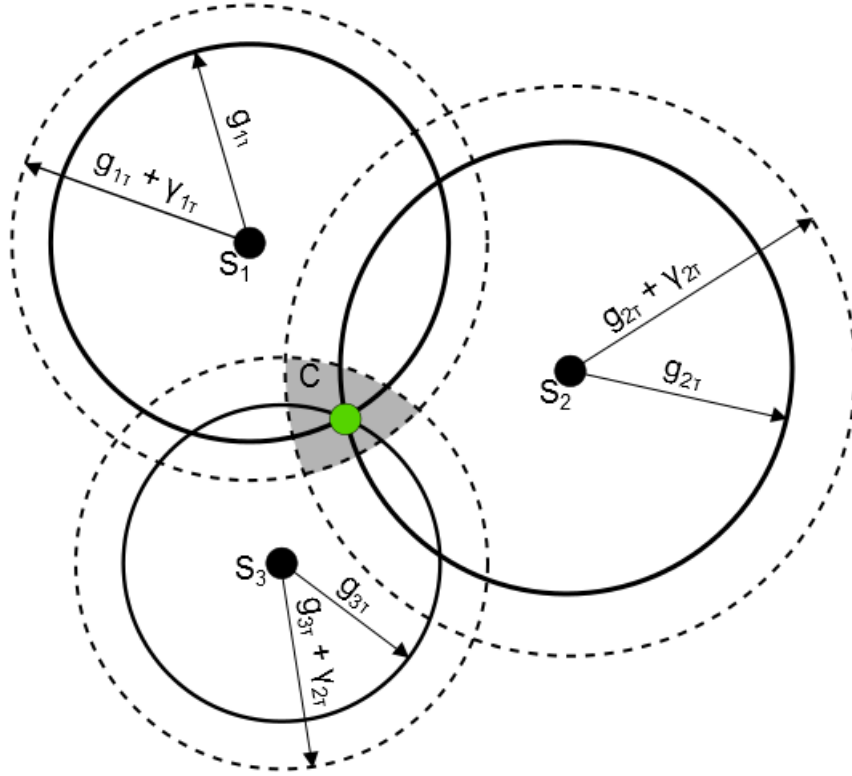
2.3.1 Multilaterace v CBG

Chceme-li zjistit neznámou polohu bodu a máme-li k dispozici několik bodů se známou polohou (pro přesné měření alespoň tři), pak jsme schopni tak učinit. Dvě základní metody k tomuto účelu hojně využívané jsou multiangulace a multilaterace, přičemž metoda CBG využívá druhou zmíněnou. Rozdíl mezi těmito metodami je v tom, jaké údaje potřebují. Pro multiangulaci je to směr (úhel), pro multilateraci je to vzdálenost od referenčních bodů. Pojem triangulace znamená určení polohy pomocí úhlů k *třem* referenčním bodům. Někdy se mylně užívá pro jiné metody určení polohy.

Základní myšlenka metody CBG vychází z metody SOI, viz kap. 2.2.2. Mějme množinu aktivních sond $S = \{S_1, S_2, \dots, S_K\}$ o velikosti K . Pro účely multilaterace je potřeba zjistit vzdálenost cíle od těchto sond. Zde vyvstává problém této metody. Vzdálenost nejsme schopni přesně změřit, pouze ji odhadujeme z naměřeného zpoždění při přenosu signálu mezi stanicemi.

Jak bylo napsáno v kap. 1, zpoždění v IP sítích se skládá ze dvou částí, deterministické (pevné) a stochastické (proměnlivé). Deterministické zpoždění se skládá z minimálních časů zpracování paketů síťovými prvky, dobou vysílání a dobou šíření signálu. Je pevně dané pro libovolnou cestu. Stochastické zpoždění tvoří čekání ve vstupních/výstupních frontách na mezilehlých směrovačích a proměnlivý čas zpracování směrovačů, který přesahuje minimální hodnotu.

Obr. 2.1 zobrazuje multilateraci v CBG s množinou sond $S = \{S_1, S_2, S_3\}$. Každá sonda S_i odvozuje svoji vzdálenost k cíli C . Tato vzdálenost se skládá ze dvou částí



Obr. 2.1: Multilaterace v CBG s omezením geografické vzdálenosti

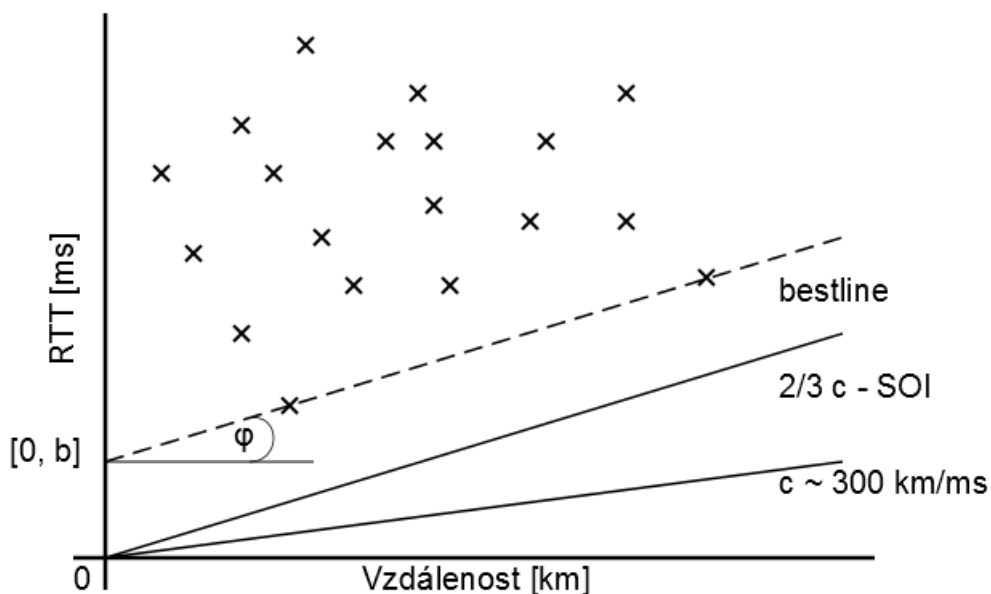
$\hat{g}_{i\tau} = g_{i\tau} + \gamma_{i\tau}$, tedy reálné geografické vzdálenosti $g_{i\tau}$ plus aditivního geografického zkreslení $\gamma_{i\tau}$. Toto čistě aditivní geografické zkreslení vzniká společně s aditivním zkreslením zpoždění při přenosu. Důsledkem toho je, že poloha cílové stanice C není jeden bod, ale oblast, na obr. 2.1 zvýrazněna šedou barvou, tato oblast je dána průnikem nadhodnocených geografických vzdáleností od sond k cílové stanici.

2.3.2 Převod zpoždění na geografickou vzdálenost

Základem přepočtu je poznatek uvedený v kap. 1 a to fakt, že rychlost šíření signálu v optickém médiu je rovna $\frac{2}{3}$ rychlosti světla ve vakuu. V [14, 17] je na základě tohoto faktu udáváno zpoždění vzniklé přenosem signálu 1 ms RTT na 100 km kabelu, na obr. 2.2 je to přímka s označením $\frac{2}{3}c$ - SOI.

Takovýmto způsobem získává geografickou vzdálenost metoda SOI. Metoda CBG tuto vzdálenost ještě omezí (zkrátí). Pro každou sondu S_i se změří zpoždění d_{ij} a geografická vzdálenost g_{ij} , kde $i \neq j$. Tyto závislosti zpoždění na vzdálenosti se vynesou do grafu jako je na obr. 2.2. Z naměřených hodnot získáme přímku *bestline*, která má parametry pro nejmenší aditivní zpoždění sondy S_i . Tato přímka musí

„ležet“ pod všemi body v grafu a její směrnice je kladná. Lze ji vyjádřit rovnicí přímky $y = a_i \cdot x + b_i$, kde a_i je směrnice přímky a b_i je bod, jímž prochází přímka na ose y. Směrnici lze spočítat ze vztahu $a = \tan(\varphi)$, kde φ je úhel, který svírá přímka s osou x.



Obr. 2.2: Příklad grafu závislosti zpoždění na geografické vzdálenosti pro jednu sondu

Všechny sondy následně používají své hodnoty a_i a b_i k spočítání geografické vzdálenosti pomocí vztahu:

$$\hat{g}_{i\tau} = \frac{d_{i\tau} - b_i}{a_i}. \quad (2.1)$$

Pokud jsou zpoždění mezi sondami periodicky měřena, pak se metoda CBG může sama kalibrovat na dynamicky se měnící podmínky v síti.

2.3.3 Zjištění polohy cílové stanice

CBG používá geometrický přístup k zjištění polohy cílové stanice. Každá sonda S_i zná svoji geografickou vzdálenost od cíle získanou pomocí vztahu 2.1. Pro každou sondu pak cíl leží uvnitř kruhové oblasti O_τ se středem v geografické poloze sondy S_i . Pro zadanou množinu sond o velikosti K získáme množinu kružnic

$O_\tau = O_{1\tau}, O_{2\tau}, \dots, O_{K\tau}$. Poloha cíle pak musí být v oblasti průniku všech kružnic. Matematicky to lze popsat vztahem:

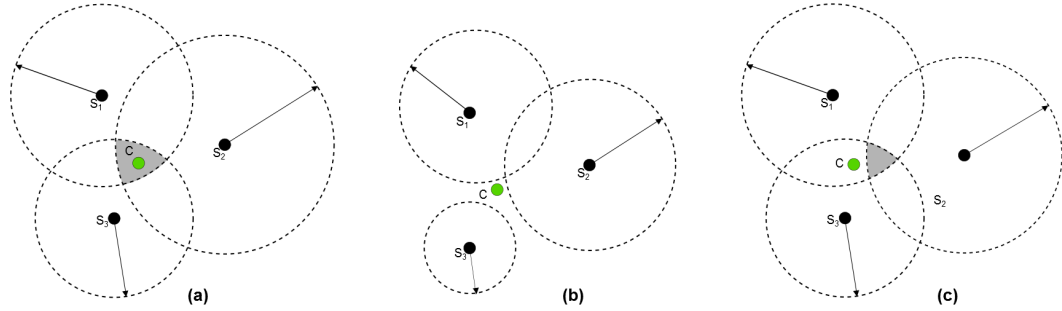
$$R = \bigcap_i^K O_{i\tau}. \quad (2.2)$$

2.3.4 Důsledky podhodnocení a nadhodnocení vzdálenosti

Při vytváření oblastí O_τ pro daný cíl C mohou nastat tři situace:

- Geografické vzdálenosti od všech sond jsou nadhodnocené.
- Geografické vzdálenosti od všech sond jsou podhodnocené.
- Pro některé sondy jsou geograf. vzdálenosti nadhodnocené, pro zbylé podhodnocené.

Tyto situace jsou zobrazeny na obr. 2.3.



Obr. 2.3: Důsledek nadhodnocených a podhodnocených geografických vzdáleností. (a) Nadhodnocené vzdálenosti. (b) Podhodnocené vzdálenosti. (c) Kombinace předchozích.

V případě zobrazeném na obr. 2.3(a), je metoda CBG schopna zjistit průnik oblastí R a odhadnout polohu cíle C . Díky charakteru zpoždění blíže popsáném v kap. 1, je toto považováno za jedinou možnou situaci, která může nastat.

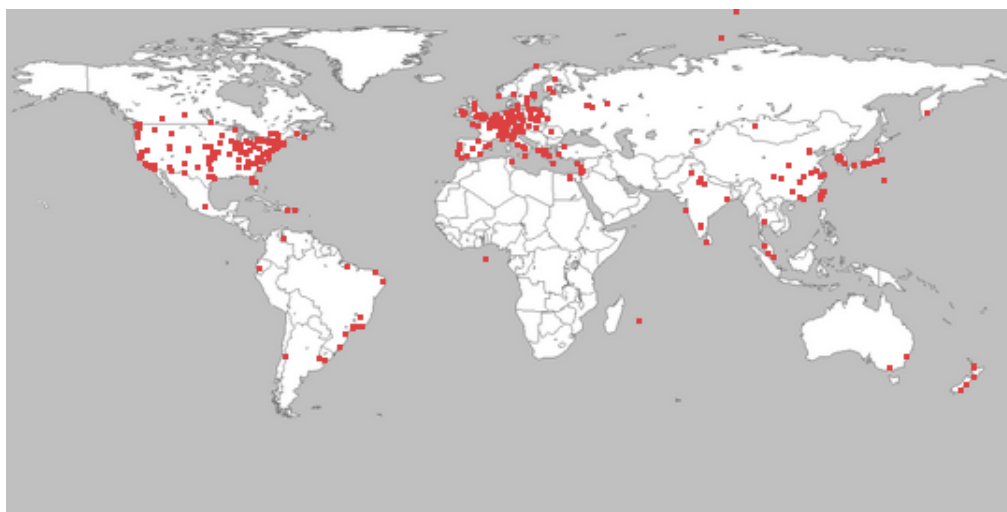
Nastane-li případ (b), pak neexistuje průnik všech oblastí R . I když nepravděpodobná, tato situace může nastat pokud má cílová stanice C menší poměr geografické vzdálenosti ku přenosovému zpoždění, než je reprezentováno přímkou *bestline* (viz 2.2), tedy menší než pro *všechny* sondy. CBG v tomto případě není schopná určit geografickou polohu.

V případě (c) sondy S_1 a S_2 svoji geografickou vzdálenost k cíli C nadhodnotili, zatímco sonda S_3 svoji vzdálenost podhodnotila. Tím vzniká situace, kdy je vytvořena oblast průniků R , ale tato neobsahuje cílovou stanici C . Výskyt této situace se stejně jako v případě (b) nepředpokládá.

3 PLATFORMA PLANETLAB

PlanetLab je otevřená platforma pro vývoj, implementaci a přístup ke službám na celosvětové úrovni. Jedná se o globální výzkumnou síť, která vytváří základ pro testování různých, převážně akademických, aplikací[18].

Síť vznikla v roce 2002 na popud Larryho Petersona (Princeton) a Davida Culle (UC Berkeley a Intel Research) na území Spojených států amerických. K 1. 12. 2012 má jen Evropská část 306 uzlů v 152 lokalitách, viz obr. 3.2, více o struktuře je v kap. 3.1. Lokality v rámci celého světa jsou vidět na obr. 3.1.



Obr. 3.1: Lokality sítě PlanetLab na mapě světa, obrázek čerpán ze zdroje[18]

Konsorcium PlanetLab je společenství akademických, průmyslových a vládních institucí, jenž spolupracují v podpoře a rozšiřování infrastruktury sítě PlanetLab. Je zodpovědné za dohled nad dlouhodobým růstem hardwarové infrastruktury, návrhem a správou softwarové architektury, poskytováním technické podpory a definováním politik správného užívání sítě. V čele konsorcia jsou univerzita Princeton, university of California at Berkeley a University of Washington. Z korporátních členů stojí za zmínku:

- Intel
- Hewlett Packard
- Google
- AT&T

Instituce, které chtějí vstoupit do konsorcia musí podepsat členskou dohodu a připojit do sítě PlanetLab alespoň dva uzly.



Obr. 3.2: Evropské lokality sítě PlanetLab, obrázek čerpán ze zdroje[19]

3.1 Struktura PlanetLab

Aby byla síť PlanetLab lépe spravovatelná, dělí se na dvě části. PlanetLab Europe (PLE), který jak název napovídá působí v Evropě a je spravován z Paříže. Pod PlanetLab Central (PLC) náleží zbytek světa.

Základem sítě jsou lokality (sites). Lokalita je geografická oblast, která obsahuje několik uzlů (nodes). Poloha lokalit v Evropě je vidět na obr. 3.2. Uzel je server, na němž běží komponenty služeb PlanetLab. Řezy (slices) jsou již alokované zdroje uživateli (user). Pro většinu uživatelů řez znamená mít přístup pomocí příkazové řádky k několika uzlům.

Řezy jsou přiřazeny určitému počtu uzlů. Když má uživatel přístup k řezu, může mu začít přiřazovat uzly. Jakmile je uzel k řezu přiřazen, je v uzlu vytvořen virtuální server. V uzlech jsou virtualizovány kopie CentOS, distribuce OS Linux, který je založen na Red Hat Enterprise Linux[3]. Každý uživatel má přístup k vlastní kopii proto, aby jednotlivé experimenty neměli možnost se navzájem ovlivnit. Tříska nebo štep je označení řezu běžícího na jednom specifickém uzlu PlanetLab.

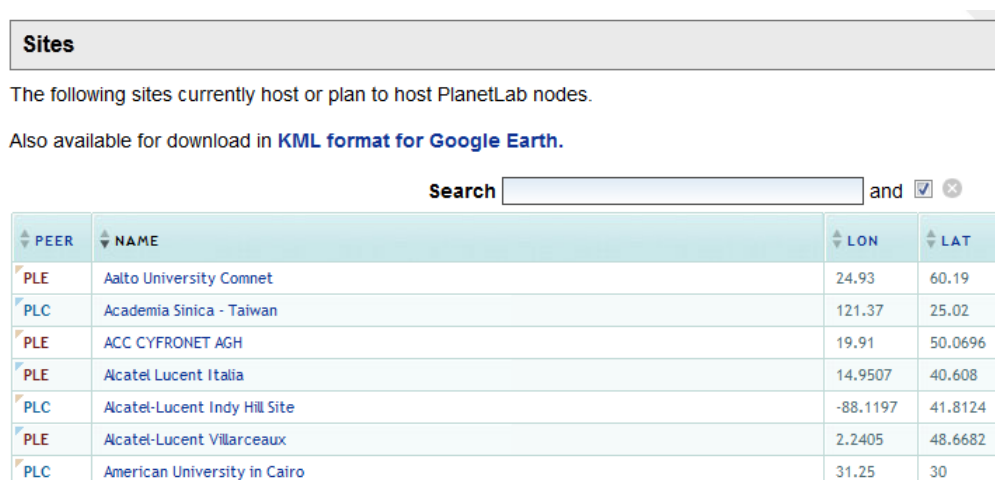
3.2 Správa serverů

Řezy je možno spravovat pomocí dvou nástrojů. Jednoduší operace lze provádět pomocí webového rozhraní. K hlubší konfiguraci se daleko více hodí použít oficiální rozhraní pro programování aplikací (Application Programming Interface – API). Vhodnou pomůckou je i dokumentace¹. Využít lze i neoficiální nástroje, které implementují některé funkce API pomocí grafického uživatelského rozhraní (Graphic User Interface – GUI).

Přístup k jednotlivým štěpům je prováděn pomocí protokolu SSH (Secure Shell). Uživatelské jméno je název řezu.

3.3 Využití PlanetLabu pro geolokaci

V kap. 2 jsou popsány principy geolokace a implementovaná metoda v rámci této práce. PlanetLab je použit proto, že jak je vidět na obr. 3.2, má v Evropě pokrytí dostatečné pro požadovanou přesnost měření. Dále má u všech lokalit uvedenu jejich přesnou geografickou polohu, což je další, velmi důležitý faktor pro to, aby mohl být použit.



Sites

The following sites currently host or plan to host PlanetLab nodes.

Also available for download in [KML format for Google Earth](#).

Search and ☒ ☐

PEER	NAME	LON	LAT
PLE	Aalto University Comnet	24.93	60.19
PLC	Academia Sinica - Taiwan	121.37	25.02
PLE	ACC CYFRONET AGH	19.91	50.0696
PLE	Alcatel Lucent Italia	14.9507	40.608
PLC	Alcatel-Lucent Indy Hill Site	-88.1197	41.8124
PLE	Alcatel-Lucent Villarcoux	2.2405	48.6682
PLC	American University in Cairo	31.25	30

Obr. 3.3: Část seznamu lokalit, čerpáno z[18]

Na obr. 3.3 je vidět prvních sedm řádků z výpisu lokalit na webových stránkách PlanetLabu. U lokality je uvedeno kam náleží, jestli pod PLC nebo PLE, jaké instituci patří a její geografická poloha.

¹ Dokumentace a mnoho dalších zdrojů informací jsou dostupné z: <<https://www.planet-lab.org/doc>> , API je popsáno na: <https://www.planet-lab.org/doc/plc_api>

4 MAPOVÉ PROJEKCE

Mapové projekce jsou metody, jejichž úkolem je reprezentace trojrozměrného tělesa v rovině. Jsou nezbytné k tvorbě map. Všechny mapové projekce zanáší určitá zkreslení povrchu. Dle funkce mapy jsou některá zkreslení akceptovatelná a jiná ne. Proto existuje poměrně velké množství metod s různými vlastnostmi.

Vlastnosti mapových projekcí mohou být vyjádřeny těmito parametry[12]:

- **Vzdálenost** – Jak přesně ji lze měřit?
- **Vzdálenost dvou bodů** – Jak jednoduše ji lze získat?
- **Směr** – Zůstal zachovaný?
- **Tvary** – Zůstaly stejné?
- **Poměr velikostí oblastí** – Zůstal zachován?
- **Zkreslení** – Jaké oblasti jsou zkresleny a jak moc?

Na základě odpovědí na výše zadané otázky, lze projekce dělit na:

- **Délkojevná** (stejnodélková, ekvidistantní) – nezkreslují vzdálenosti dle nějakého systému čar. Obvykle to neplatí všemi směry.
- **Plochojevná** (stejnoplochá, ekvivalentní) – Zachovává poměry ploch, zkresluje úhly.
- **Úhlojevná** (stejnoúhlá, konformní) – Věrně zachycují úhly, zkreslují plochy.
- **Vyrovňovací** (kompenzační) – Kompromis mezi výše zmíněnými, zkresluje plochy i úhly.

4.1 Válcové projekce

Termín válcová projekce je použitelný pro všechny projekce, ve kterých jsou poledníky namapovány na stejně vzdálené vertikální čáry a rovnoběžky jsou čáry horizontální[16].

Mapování poledníků na vertikální čáry si lze jednoduše představit, vezmeme-li válec, jehož osa je totožná se světovou osou rotace. Tento válec pak zemi obalí, přičemž se na něj „přenesou povrch země“. Následně se odroluje.

Většina válcových projekcí mapuje celou Zemi jako obdélník konečných rozměrů. Existují však výjimky, při nichž je obdélník nekonečně vysoký při konstantní šířce.

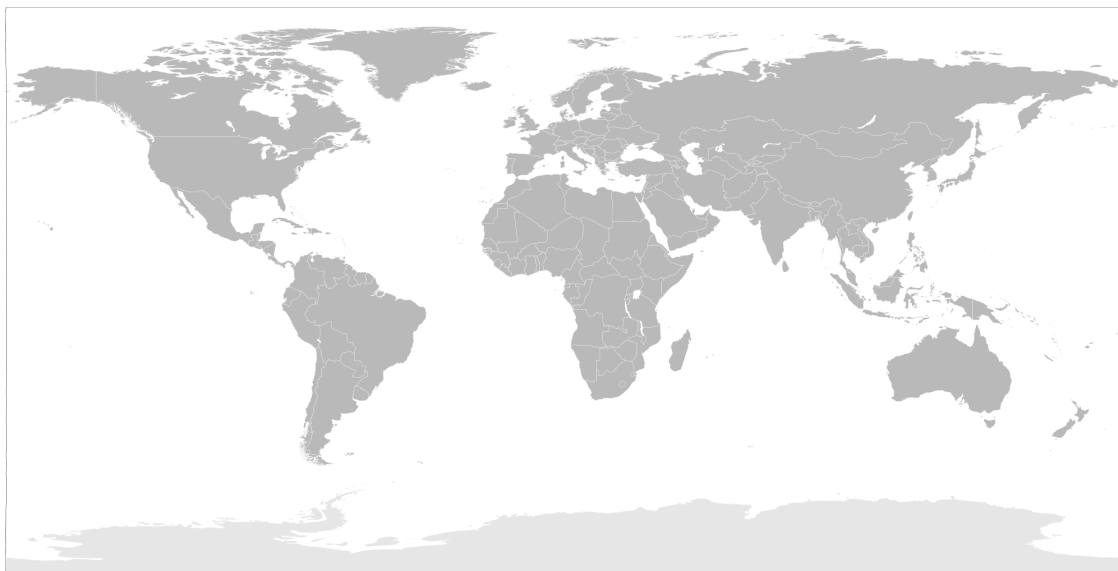
Válcová projekce je využita například v mapách google¹, a ekvidistantní válcová projekce byla použita při řešení této práce.

¹ Jedná se o tzv. Mercatorovu projekci, samy mapy jsou dostupné z: <<https://maps.google.com/>>

4.2 Ekvidistantní válcová projekce

Ekvidistantní válcová projekce, dále jen EVP, byla vybrána ze dvou důvodů. Zaprvé je jejím výstupem spojitá plocha obdélníkového tvaru, což je pro převod do kartézských souřadnic velká výhoda. Pro tento účel by byly některé jiné druhy projekcí, například pseudokónické² velmi nevhodné. Zadruhé je tato projekce ve směru poledníků délkojevná[12]. EVP není úhlojevná ani plochojevná, to však není pro účely této práce podstatné.

V EVP jsou poledníky rovnoměrně rozmístěné, stejně dlouhé vertikální čáry. Totéž platí i pro rovnoběžky, což je fakt, který musí být při samotné implementaci zohledněn. Ukázková mapa vytvořená EVP je na obr. 4.1. Na tomto obrázku je taky vidět zkreslení, které tato metoda zanáší ve směru osy x. Na rovníku, je toto zkreslení nulové a směrem k pólům roste. Proto se například Grónsko jeví větší než Austrálie, i když je tomu naopak.



Obr. 4.1: Slepá mapa světa vytvořená EVP

$$x = R \cdot \lambda \cdot \cos \varphi_0 \quad (4.1)$$

$$y = R \cdot \varphi \quad (4.2)$$

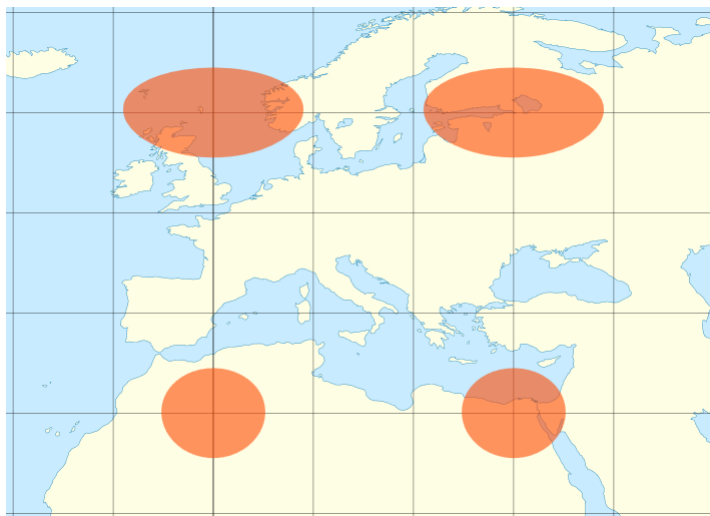
Bod, u něž známe jeho polohu udanou pomocí sférických souřadnic (Severní/-Jižní šířka, Východní/Západní délka), jsme schopni pak pomocí EVP převést do kartézských souřadnic pomocí vztahů 4.1 a 4.2, kde x je horizontální souřadnice

²příklad dostupný z: <<http://www.progonos.com/furuti/MapProj/Normal/ProjPCon/projPCon.html>>

bodů, y je vertikální souřadnice bodu, R^3 je poloměr Země, λ je zeměpisná délka, φ je zeměpisná šířka a φ_0 jsou rovnoběžky na sever i jih od rovníku, kde není zkreslení ve směru osy x . Na φ_0 závisí i poměr stran výsledného obdélníku, zvolíme-li $\varphi_0=0$, pak má obdélník poměr stran 2:1 a osa x nemá žádné zkreslení na rovníku. Více je k problematice zkreslení EVP a jeho vlivu na řešení této práce napsáno v následující podkapitole.

4.3 Vliv EVP na tvar oblasti

Na obr. 2.1 jsou zobrazeny oblasti možného výskytu cíle jako kružnice. V předchozí kapitole je pro zvolenou metodu zobrazení použitou v této práci napsáno, že ve směru poledníků si měřítko zachovává, zatímco ve směru rovnoběžek zkreslení s rostoucí zeměpisnou šířkou roste. Na obr. 4.2 je vidět takzvaná Tissotova elipsa, která ukazuje, jak je v daném místě zkreslen kruh určitého průměru. Jak je z onoho obrázku patrné, oblast je přibližně, ne však úplně eliptického tvaru. Jak se mění zkreslení v závislosti na změně zeměpisné šířky je pro vybrané hodnoty vypsáno v tab. 4.1.



Obr. 4.2: Závislost velikosti zkreslení na zvětšující se vzdálenosti od rovníku, čerpáno z [11]

Zkusme nahradit tuto zkreslenou elipsu elipsou normální. Vezměme několik bodů, které jsou na zemské kouli stejně vzdálené od jednoho bodu (tvoří přibližně kružnici na povrchu Země). Pro tento účel byly použity mapy Google. Střed byl zvolen na 40° severní šířky a 20° východní délky. Od tohoto bodu, dále nazývaného jako střed, bylo v půlkruhu o poloměru 1500 km naměřeno osm dalších bodů. Jejich sférické souřadnice ve stupních byly nejdříve převedeny na radiány, následně pomocí vztahů

³ v této práci bude předpokládáno, že země je koule o poloměru 6371 km

4.1 a 4.2 převedeny na souřadnice v soustavě os x a y . Rovnice elipsy v kanonickém tvaru[8] je:

$$1 = \frac{(x - x_0)^2}{a^2} + \frac{(y - y_0)^2}{b^2}. \quad (4.3)$$

Vstupní proměnné x a y jsou souřadnice libovolného bodu ležícího na obvodu elipsy. Střed elipsy leží v bodě x_0 a y_0 . Rozměr elipsy je dán délkou hlavní poloosy a a vedlejší poloosy b . Pravá strana rovnice je rovna jedné, pokud bod leží právě na obvodu. Pokud bod leží uvnitř elipsy, pak je pravá strana menší než jedna, je-li bod vně elipsy, pak je pravá strana větší než jedna.

Zeměpisná šířka [°]	Zkreslení [-]
0	1.000
10	1.015
20	1.064
30	1.154
40	1.305
50	1.555
60	1.999
65	2.366
70	2.923
75	3.863
80	5.758
85	11.473
90	∞

Tab. 4.1: Zkreslení EVP na různých zeměpisných šířkách

V tabulce 4.2 je v prvních dvou sloupcích geografická poloha bodu udaná sférickými souřadnicemi. Ve třetím a čtvrtém sloupci jsou tyto hodnoty převedeny na hodnoty x a y . V posledních dvou sloupcích jsou tyto hodnoty dosazeny do rov. rovRovniceElipsy. Délka hlavní poloosy a je spočítána jako poloměr vynásobený zkreslením na dané zeměpisné šířce. Zkreslení jsem získal jako podíl obvodu rovníku ku obvodu rovnoběžky, na které leží daná poloosa. Délka poloosy b je rovna poloměru kruhu.

Jak je vidět z předposledního sloupce, některé naměřené body leží mimo sestrojenou elipsu. Proto byly délky poloos experimentálně zvětšovány, dokud všechny body neležely uvnitř elipsy. Tomu se tak stalo až poté, co byly délky zvětšeny o 4,5%.

Operace umělého zvětšování poloměru se může ve výsledné přesnosti geolokační metody odrazit dvěma způsoby – může vést ke zvětšení nepřesnosti metody CBG a může napravovat případné podhodnocené vzdálenosti.

Sférické souřadnice		Kartézské souřadnice		Dosazení do rovnice elipsy	
Sev. šíř. [°]	Vých. dél. [°]	x	y	bez korekce	s korekcí
40.00	20.00	2223.898	4447.797	0 (střed)	0 (střed)
26.55	20.00	2223.898	2952.225	0.994	0.910
27.70	26.10	2902.187	3080.099	0.951	0.871
30.63	31.72	3527.103	3405.900	0.925	0.847
37.00	36.75	4086.413	4114.212	0.954	0.873
40.00	37.62	4183.153	4447.797	1.001	0.916
43.00	37.58	4178.705	4781.381	1.046	0.957
50.81	31.69	3523.767	5649.814	1.082	0.991
53.02	25.47	2832.134	5895.555	1.028	0.941
53.56	20.02	2226.122	5955.600	1.010	0.925

Tab. 4.2: Poloha bodů udaná sférickými a ortogonálními souřadnicemi a jejich dosazení do rovnice elipsy.

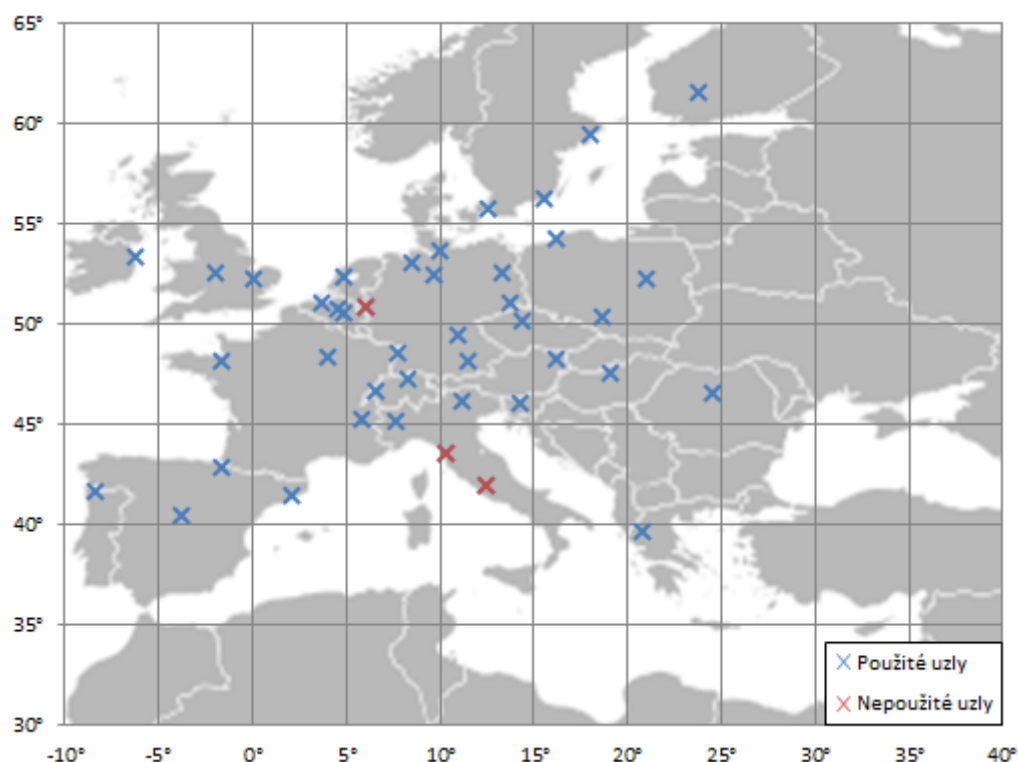
5 NÁVRH ŘEŠENÍ A IMPLEMENTACE METODY CBG

Hlavním cílem této semestrální práce je vytvořit program, jenž implementuje metodu geolokační metodu CBG a následně ověřit jeho funkčnost. V této kapitole bude popsán postup získání dat pro geolokaci a vlastní implementace pomocí programovacího jazyka JAVA. Při tom bude navazováno na informace udané v předchozích kapitolách.

5.1 Měření dat pro Geolokaci

Pro geolokaci pomocí metody CBG, jak je napsané v kap.2.3, je zapotřebí mít změřené přenosové zpoždění od stanic se známou polohou ke stanici již chceme lokalizovat. Jak byla tato data získána a zpracována je vysvětleno v následujících podkapitolách.

5.1.1 Vybrané uzly Planetlabu



Obr. 5.1: Uzly PlanetLab použité pro získání dat

Před samotným měřením bylo důležité vybrat množinu uzlů sítě Planetlab. Požadavek na výběr těchto uzlů je hlavně jejich co nejrovnoměrnější rozložení měřenou oblastí (Evropou). Počet uzlů měl být srovnatelný s počtem zvoleným v[14] (zde použili 42 uzlů), aby bylo možné výsledky vzájemně porovnat.

Bylo vybráno 42 uzlů, z nichž se bylo možné v době měření připojit na 39. Poloha těchto uzlů je vidět na obr. 5.1. Seznam je uveden v tab. B.1 i na přiloženém médiu.

K připojení na uzly Planetlabu byl použit program `putty.exe`. Následně bylo potřeba vygenerovat dvojici soukromého a veřejného klíče. K tomu účelu posloužil program `puttygen.exe`. Komunikace s uzlem probíhala pomocí protokolu SSH. Po připojení byl pomocí programu `pscp.exe`¹ do uzlu nahrán Bash skript, jehož funkce je popsána níže, a textové soubory sloužící pro uložení naměřených dat. Po ukončení měření byl použit stejný program ke stažení textových souborů ze serveru.

5.1.2 Bash skripty a třídění dat

Na obr. 5.2 je vidět zjednodušený vývojový diagram Bash skriptu² nahrávaného do uzlů Planetlabu. Zjednodušeného proto, že to co je na obrázku zobrazeno se provede dvakrát. Jednou se použijí adresy uzlů planetlabu za účelem kalibrace geolokační metody, podruhé se použijí adresy uzlů, jenž mají být lokalizovány. Před samotným zpuštěním skriptu je však potřeba ho editovat a na pátém řádku přepsat proměnnou *MOJEIP* z hodnoty *X.X.X.X* na IP adresu uzlu, na němž se skript spouští.

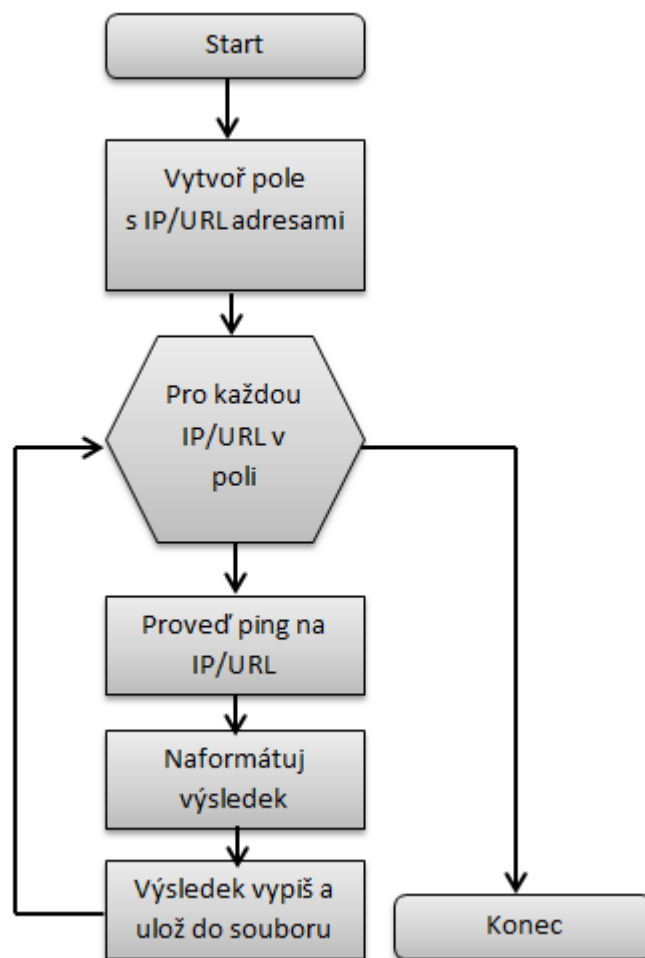
Po spuštění skriptu se postupně spustí program PING pro obě dvě skupiny adres. Získané výstupy se pomocí příkazu `grep` vyfiltrují a takto zpracovaná data se uloží do textových souborů. Data s naměřeným obousměrným zpožděním k serverům Planetlab se uloží do souboru *StS.txt*, pro uložení zpoždění k lokalizovaným stanicím slouží soubor *StC.txt*. Skript je napsán tak, aby se naměřená data přidala na konec souborů *StS.txt* a *StC.txt*, pokud se nakopírují na měřicí uzel společně se skriptem Bash. V opačném případě se při spuštění skriptu vytvoří nové.

Tato surová naměřená data je ještě potřeba upravit. Vzhledem k tomu, že skript provádí PING na všechny používané servery, provádí PING i na sebe sama a tato hodnota by zkreslila kalibrační mechanismus. Proto je potřeba z výsledné množiny výsledků vyřadit všechny, jejichž naměřená hodnota obousměrného zpoždění je menší než 0,1 ms. Další situací, již bylo potřeba ošetřit bylo zakázání odesílání ICMP paketů na cílové stanici (cílová stanice neodpovídá na PING). Vzhledem k tomu, že tyto stanice neměly pro účely této práce využití, byly ze seznamu vyřazeny.

Kompletní seznam stanic je uveden v tab. C.1.

¹Programy `putty.exe`, `puttygen.exe` a `pscp.exe` jsou dostupné z: <<http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html>>.

²Tento skript je možné najít na přiloženém médiu pod názvem `PingFinal.sh`



Obr. 5.2: Zjednodušený vývojový diagram skriptu Bash pro získání dat

5.1.3 Soubory s naměřenými daty

V této podkapitole budou popsány čtyři textové soubory, s nimiž vyvinutý program pracuje.

Dva z nich, *BStS.txt* a *BStC.txt* obsahují kompletní naměřená data, ze kterých jsou odebrány nesmyslné údaje popsané v kap. 5.1.2. Oba mají stejnou strukturu, jeden záznam odpovídá jednomu řádku, tato logika zůstává zachována u všech souborů popsaných v této kapitole, přičemž na každém z řádků jsou data ve formátu: <IP adresa cíle PINGu> <IP adresa zdroje PINGu> <naměřené obousměrné zpoždění>. Prvních pět řádků ze souboru *BStC.txt* je k vidění na obr. 5.3.

Soubor *seznam_stredu_complete.txt* obsahuje údaje o uzlech sítě Planetlab. Struktura záznamu je následující: <IP adresa uzlu> <Poloha - Východní délka [°]> <Poloha - Severní šířka [°]> <a> . Význam parametrů <a> a je vysvětlen v kap. 2.3.2. Ilustrační obsah souboru je k vidění na obr. 5.4.

```

1 176.9.55.42 131.130.69.162 11.180
2 91.82.84.185 131.130.69.162 27.593
3 83.222.115.86 131.130.69.162 69.110
4 80.72.40.110 131.130.69.162 20.223
5 109.70.149.49 131.130.69.162 32.038

```

Obr. 5.3: Obsah souboru s naměřenými daty

```

1 131.130.69.162 16.24 48.16 0.020024265633916533 0.2530559794047651
2 195.116.53.24 21.0069 52.2221 0.019444671220457822 3.5193717194037033
3 130.104.72.213 4.61667 50.6833 0.01694102725123346 5.7388331783995605
4 128.232.103.201 0.0917 52.2112 0.019102725290236117 3.179668208728623
5 131.254.208.10 -1.65071 48.1155 0.022514959428783572 0.05839756974002697

```

Obr. 5.4: Obsah souboru *seznam_stredy_complete.txt*

Posledním souborem využívaným v programu je *Originalni_pozice.txt*. Zde jsou uloženy záznamy o pozici stanic jejichž poloha byla měřena. Data jsou ve formátu: <IP adresa stanice> <Poloha - Východní délka [°]> <Poloha - Severní šířka [°]>. Ilustrační obsah souboru je k vidění na obr. 5.5.

```

1 176.9.55.42 10.451526 51.165691
2 91.82.84.185 19.040758 47.498406
3 77.251.170.54 5.138279 51.733004
4 83.222.115.86 37.618422 55.751242
5 80.72.40.110 21.012229 52.229676

```

Obr. 5.5: Obsah souboru *Originalni_pozice.txt*

5.2 Algoritmus pro zjištění nejvhodnější přímky

V této kapitole je vysvětleno získání kalibračních dat teoreticky popsané v kap. 2.3.2. Budou popsány jednoduché matematické principy stojící za výběrem nejlepší přímky a následně i samotná implementace v programu.

5.2.1 Matematika na pozadí

Tato podkapitola se věnuje geometrickým výpočtům provázejících problematiku rovnice přímky ve dvourozměrném prostoru a polohu bodů relativně k ní.

Přímku procházející body $B_1 [x_1, y_1]$ a $B_2 [x_2, y_2]$ sestrojíme tak, že dosadíme oba body do rovnice přímky. Tím dostaneme soustavu dvou rovnic o dvou neznámých

$$y_1 = a \cdot x_1 + b$$

$$y_2 = a \cdot x_2 + b,$$

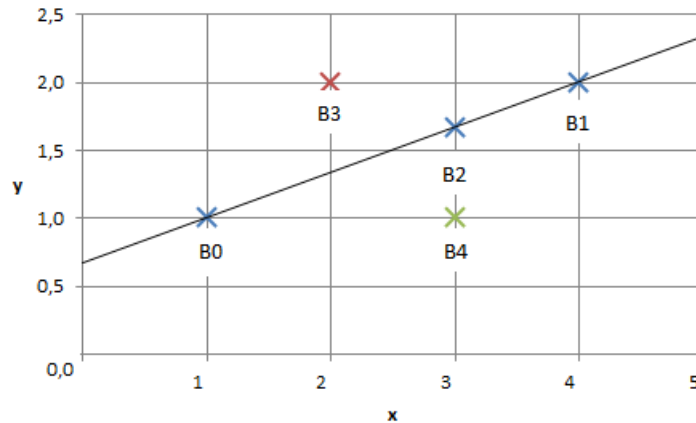
z čehož lze vyjádřit

$$a = \frac{y_1 - y_2}{x_1 - x_2}$$

a

$$b = y_1 - a \cdot x_1.$$

Proložme přímkou body $B_0 [1; 1]$ a $B_1 [4; 2]$. Rovnice této přímky je $y = x/3 + 2/3$. Vezmeme-li bod $B_2 [3; 5/3]$, pak rovnice přímky bude mít levou (dále v textu označovaná jako L) i pravou stranu (dále v textu označovaná jako P) stejnou, což znamená, že daný bod leží na této přímce. Zkusíme-li dosadit do rovnice bod $B_3 [2; 2]$, který leží „nad přímkou“, vyjde $2 = 2/3 + 2/3 = 4/3$, tedy ($L > P$). Analogicky když dosadíme do rovnice přímky bod, který leží „pod přímkou“, třeba $B_4 [3; 1]$, vyjde $1 = 3/3 + 2/3 = 5/3$, ($L < P$). Vše je zobrazeno na obr. 5.6.



Obr. 5.6: Body B0–4 zobrazené v kartézské soustavě souřadnic

5.2.2 Implementace v programu

Samotný algoritmus získání nejlepší přímky lze rozdělit do dvou částí, přípravu naměřených dat a samotný výpočet kýžené přímky. Není přímou součástí hlavního programu, je realizován jako samostatný krátký program a nachází se v balíčku *smernice*. V rámci této kapitoly, pokud bude zmíněn program a nebude řečeno jinak, je tím myšlen program na počítání rovnic přímky a výběr nejvhodnější z nich.

V kap. 5.1.3 je popsán soubor *seznam_stredy_complete.txt*. V tuto chvíli ještě nejsou koeficienty $\langle a \rangle$ a $\langle b \rangle$ spočítány. Tento program pracuje se souborem se stejnou strukturou, jen o tyto dva parametry ochuzenou *SeznamServeruSOI.txt*. Dalším souborem, jenž je použit je soubor s naměřenými zpožděními mezi uzly Planetlab *StS.txt*.

Funkce algoritmu v první části je poměrně triviální a nebylo považováno za nutné pro ni dělat vývojový diagram. Postupně se prochází soubor s naměřenými daty a pro všechny použité uzly se vytvoří pole, jenž má v každém prvku další pole o dvou prvcích. Zde je jedním prvkem naměřené přenosové zpoždění ke všem ostatním uzlům v milisekundách a druhým vzdálenost uzlů vypočítaná pomocí vztahu

$$d = R \cdot c \quad (5.1)$$

(tzv. rovnice harvesine[5]) v km, kde c spočítáme pomocí vztahu

$$c = 2 \cdot \operatorname{atan2} \left(\sqrt{a}, \sqrt{(1-a)} \right) \quad (5.2)$$

a a získáme dosazením do

$$a = \sin^2 (\Delta\varphi/2) + \cos (\varphi_1) \cdot \cos (\varphi_2) \cdot \sin^2 (\Delta\lambda/2). \quad (5.3)$$

$\operatorname{Atan2}^3$ je funkce arkus tangens, používaná v programovacích jazycích, se dvěma vstupními parametry. Ty jsou vyžadovány kvůli správnému určení kvadrantu vypočteného úhlu, což není s pouze jedním vstupním argumentem možné[1]. V rov. 5.3 je φ_1 zeměpisná šířka prvního bodu, φ_2 druhého bodu a $\Delta\varphi$ jejich rozdíl. Analogicky $\Delta\lambda$ je rozdílem zeměpisných délek.

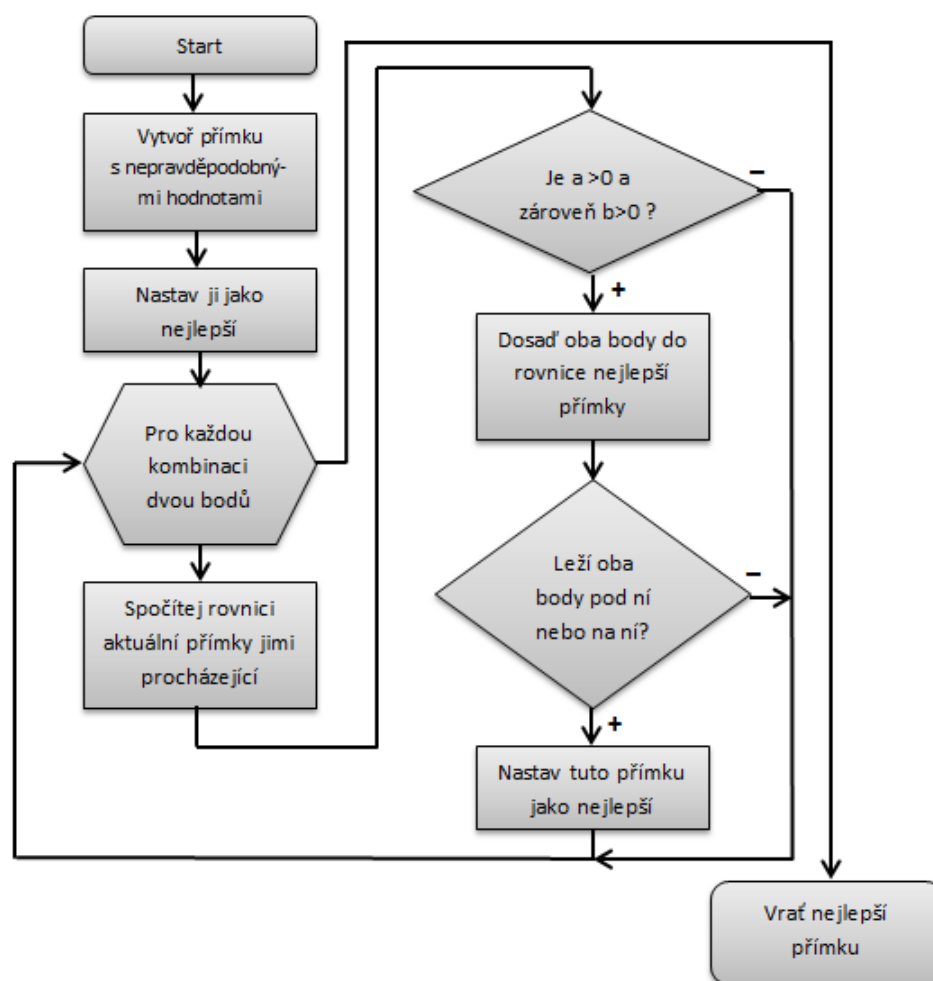
V tuto chvíli je již známo vše potřebné pro nalezení nejlepší přímky, resp. jejich koeficientů $\langle a \rangle$ a $\langle b \rangle$. Funkce algoritmu by měla být zřejmá z obr. 5.7.

Vytvořením přímky s nesmyslnými hodnotami je myšleno vytvoření takové přímky, jenž se s velkou pravděpodobností nevyskytnou mezi naměřenými daty a zároveň na těmito daty leží. Pro účely této práce byla zvolena přímka procházející body $N_1 [0; 500]$ a $N_2 [3000; 3000]$, kde první souřadnice je vzdálenost v kilometrech (na ose x) a druhá souřadnice je přenosové zpoždění v milisekundách. Tato přímka je nastavena jako nejlepší, tzv. bestline.

Pokud by z nějakého důvodu nebylo možno vybrat nejlepší přímku, pak by se použila tato přímka a program by vracel nesmyslné výsledky. Toto je ošetřeno tak, že pokud by se měla použít přímka s parametrem $\langle b \rangle$ rovným 500 (přímka prochází bodem N_1), pak se dosadí do vztahu 2.1 $b_i = 0$ a $a_i = 1$. Tedy místo metody CBG se použije metoda SOI. K tomuto během řešení práce ovšem nedošlo.

Následně program vytvoří pro každý uzel postupně všechny možné dvojice bodů vzdálenost–zpoždění a proloží jimi přímku. Má-li tato přímka jeden z koeficientů $\langle a \rangle$ či $\langle b \rangle$ menší než 0, pak se předčasně přejde na další iteraci cyklu. Není-li tomu tak, zkontroluje se, zda-li oba aktuálně používané body leží pod současnou nejlepší přímkou (viz kap. 5.2.1). Je-li splněna i tato podmínka, pak přímka procházející těmito body má správné geometrické vlastnosti a zároveň leží pod aktuálně nejlepší přímkou, je tedy nastavena jako nejlepší pro další iterace cyklu.

³Detailní popis v API je dostupný z <http://docs.oracle.com/javase/6/docs/api/java/lang/Math.html#atan2%28double,%20double%29>.



Obr. 5.7: Vývojový diagram algoritmu výběru nejlepší přímky

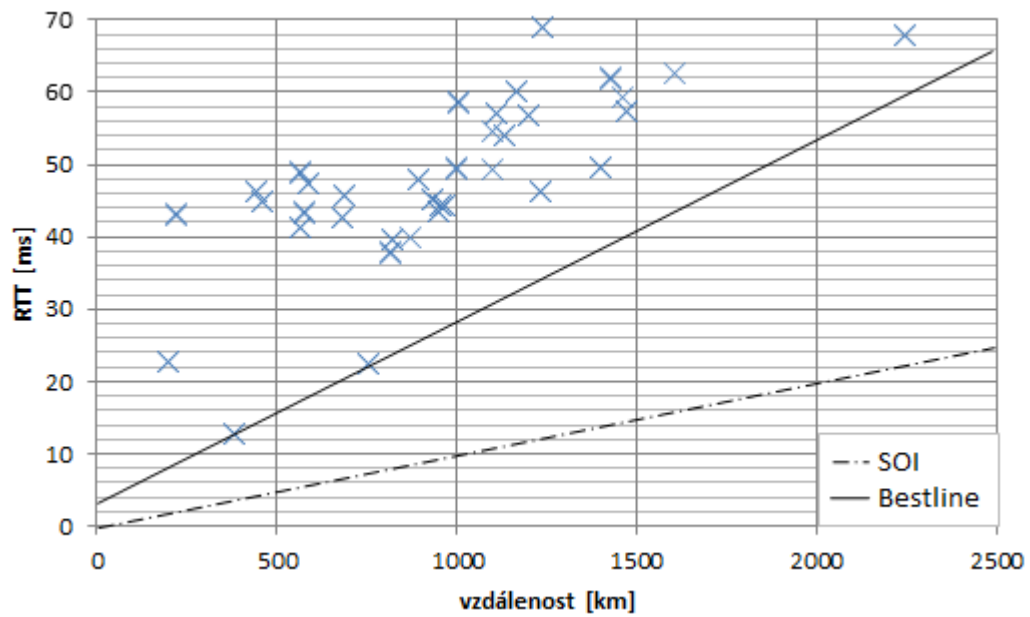
Na obr. 5.8 je vidět graficky zpracovaný výstup programu pro uzel Planetlab s IP adresou 194.47.148.170. Mimo jiné je z něj patrné, že přímka procházející body N_1 a N_2 je zvolena s dostatečnou rezervou oproti naměřeným hodnotám.

5.3 Hlavní program

V této kapitole je popsán program, jenž byl vytvořen v rámci této práce jako implementace geolokační metody CBG. Tento program se nachází v balíčku *semestrální-Projekt*.

5.3.1 Třídy

Pro účely implementace CBG byly v programu vytvořeny dvě třídy, které symbolizují prvky potřebné pro multilateraci (popsané v kap. 2.3.1). Jsou to třídy *Cíl*,



Obr. 5.8: Graficky zpracovaný výstup algoritmu programu pro jednu stanici (IP: 194.47.148.170, $a \approx 0,026$, $b \approx 2.915$)

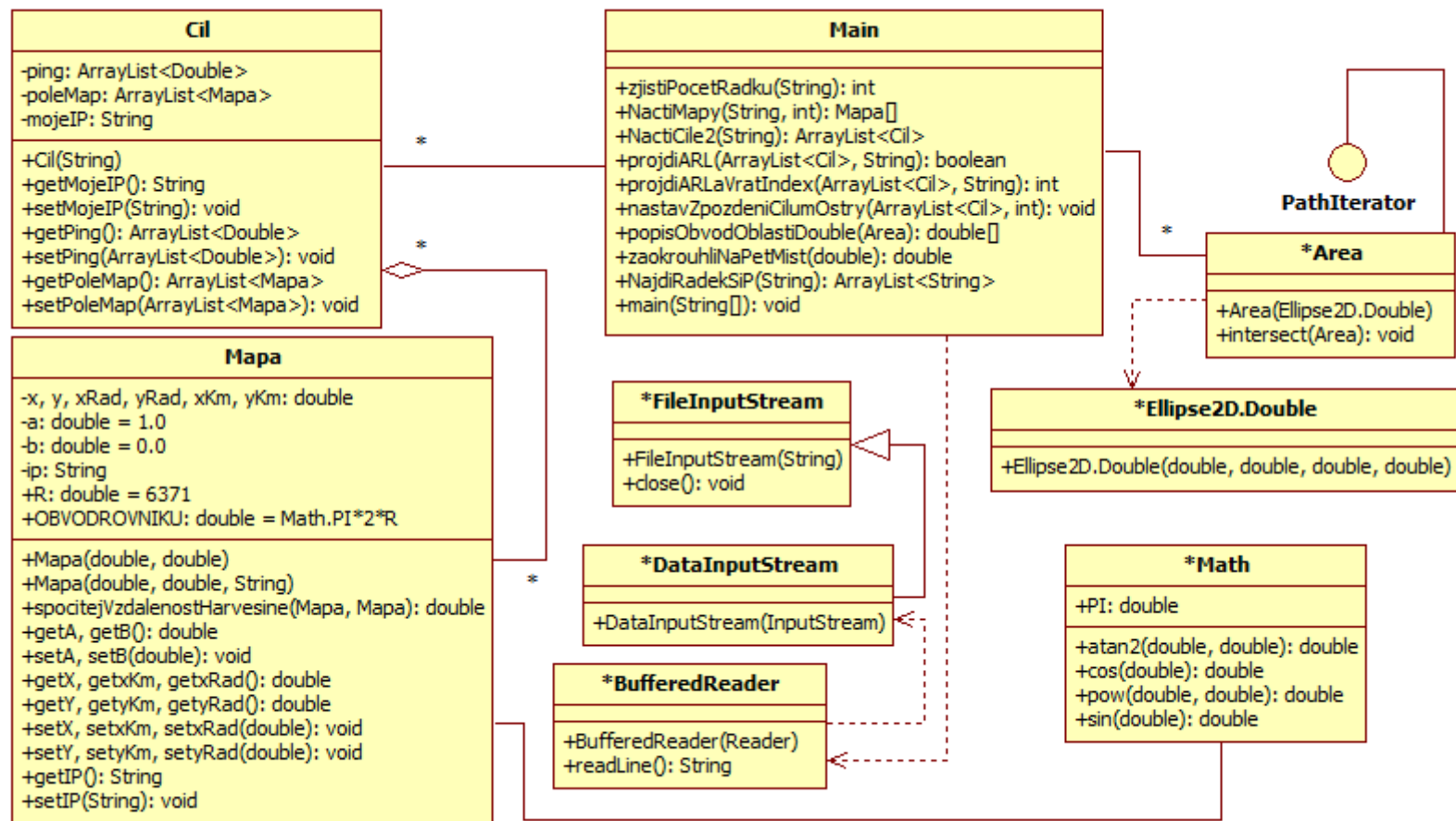
kterážto zastupuje stanici, jejíž polohu určujeme a *Mapa*, jenž zastupuje sondy (uzly Planetlabu). Účel třídy *Mapa* byl v úvodních fázích psaní programu odlišný od její finální implementace, proto nemá „logičtější“ jméno. Obě třídy jsou popsány níže. Dále je použito několik tříd, jenž jsou součástí jazyka JAVA a jejichž popis je dostupný v API. Ty v této práci budou zmíněny pouze okrajově se zaměřením na jejich stěžejní funkce (metody) pro funkci výsledného programu.

Zvlášť stojí třída *Main*. Je v ní naprogramována veškerá funkcionality implementované CBG. Vzájemný vztah vytvořených a použitých tříd je popsán v UML digramu, viz obr. 5.9.

Třídy *Cil* a *Mapa*

Instance třídy *Mapa* jsou identifikovány instanční proměnnou *ip* typu String. Poloha je uložena v proměnných *x* (zeměpisná délka) a *y* (šířka), obě jsou zadávány ve stupních jako desetinných číslech. Proměnné *xRad* a *yRad* jsou *x* a *y* převedené na radiány, *xKm* a *yKm* jsou pak tyto souřadnice v radiánech dosazené do vztahů 4.1 a 4.2.

Proměnným *a* a *b* je věnována kap. 5.2. Dokud se jejich hodnota při běhu programu nenastaví jinak, použijí se hodnoty $a = 1$ a $b = 0$.



Obr. 5.9: UML diagram tříd hlavního programu

Posledními dvěma proměnnými jsou statické R a $OBVODROVNIKU$, kde R je poloměr země a $OBVODROVNIKU$ je získán jako $2 \cdot \Pi \cdot R$.

Většina metod této třídy jsou getery a setery pracující s výše popsanými proměnnými. Mimo ně je tu metoda *spositejVzdalenostHarvesine*, jenž spočte vzdálenost dvou bodů na Zemi pomocí rovnice Harvesine (vztahy 5.1, 5.2 a 5.3).

Instance třídy *Cil* je stejně jako objekt typu *Mapa* identifikována IP adresou uloženou v proměnné *mojeIP*. Dále si drží odkazy na všechny sondy, z nichž na ni proběhlo měření RTT v poli *poleMap*. Samotná hodnota obousměrného zpoždění je uložena v poli *ping* na korespondující pozici se sondou, z níž měření proběhlo. Obě tato pole jsou dynamická, jedná se o objekty typu *ArrayList*.

Metodami třídy cíl jsou, krom jednoho konstruktoru, pouze getery a setery.

Třídy označené * na obr. 5.9

Jedná se o třídy, jenž jsou zabudované v jazyce JAVA. V programu jsou využívány pro tři skupiny operací. Pro načítání dat jsou použity třídy *FileInputStream*, *DataInputStream* a *BufferedReader*. Převážně používaná pro její goniometrické funkce, ale i pro jiné výpočty je třída *Math*. Pro modelování oblastí, výpočtu průniků a určení jejich středů jsou použity třídy *Ellipse2D.Double* a *Area* implementující rozhraní *PathIterator*.

Třída Main

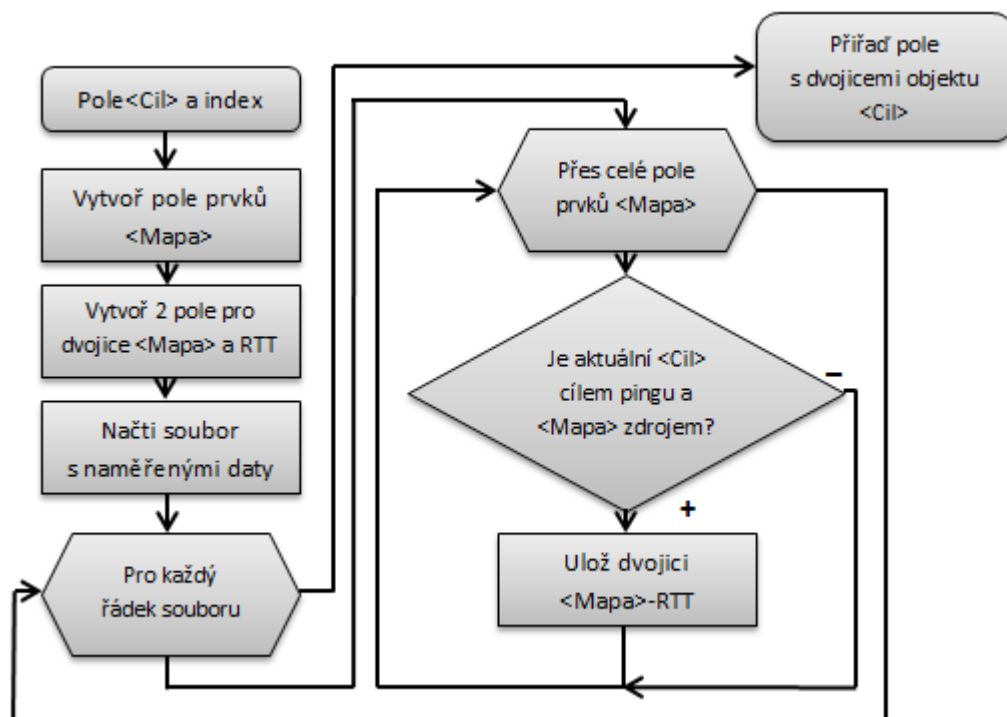
Metody, které obsahuje třída *Main* lze opět dle jejich funkce rozdělit do několika skupin. Jsou to práce se soubory, práce s poli a určování cílové polohy. Až na třídu *Math* jsou v ní použity všechny třídy popsané v předchozím odstavci.

Metody pracující se soubory jsou *zjistiPocetRadku*, *NactiMapy* a *NactiCile2*. Jejich funkci stačí popsat stručně. Metoda nazvaná výmluvně *zjistiPocetRadku* vrátí počet řádků specifikovaného souboru (platí jeden řádek—jeden záznam, viz kap. 5.1.3). Je toho využito pro určení velikosti pole objektů typu *Mapa*, kterých je přesně známý neměnný počet. Metody *NactiMapy* a *NactiCile2* vytvoří pole objektů *Mapa*, resp. *Cil*, se kterými se dále pracuje v jiných metodách.

Do skupiny práce s poli lze zařadit metody *projdiARL*, *projdiARLaVratIndex*, *nastavZpozdeniCilumOstry* a *NajdiRadekSiP*.

Metody *projdiARL* a *projdiARLaVratIndex* prochází dynamické pole s prvky typu *Mapa* a vyhledávají prvek se zadanou IP adresou (instanční proměnná *mojeip*). První pouze vrací logickou hodnotu, pokud najde shodu u nějakého prvku. Druhá metoda vrací index prvního prvku se shodnou IP adresou.

Jednou z metod, o niž se hlavně opírá funkcionality programu je metoda *nastavZpozdeniCilumOstry*. Proto zde bude podrobněji popsána. Funkce této metody



Obr. 5.10: Vývojový diagram metody *nastavZpozdeniCilumOstry*

je graficky vyobrazena na vývojovém diagramu viz obr.5.10. Jako vstupní argumenty dostane pole prvků *Cil* a index prvku, u něž se budou nastavovat instanční proměnné *ping* a *poleMap*. Nejdříve se vytvoří pole prvků typu *Mapa*, jenž mají nastaveny instanční proměnné dle serverů Planetlab (poloha, koeficienty <a> a). Následně se vytvoří dvě prázdná pole. Jedno pro prvky typu *double*, do něj se budou ukládat hodnoty naměřeného zpoždění, druhé pro prvky typu *Mapa*. Dále se načte soubor s naměřenými daty a postupně se prochází všechny záznamy. Narazí-li se na záznam, kde má cíl PINGu stejnou IP adresu jako aktuální nastavovaný objekt *Cil*, pak se projde pole s načtenými prvky *Mapa* a najde se v něm zdroj. Pak se zdroj PINGu a naměřené zpoždění uloží do dvou polí zmíněných výše tak, aby korespondující prvky byly na stejné pozici. Po projití celého souboru s naměřenými daty se tato dvě pole nastaví jako proměnné *ping* a *poleMap* objektu *Cil*.

Metoda *NajdiRadekSiP* (mimo to, že se prohřešuje proti konvencím pojmenovávání metod) slouží k načtení souboru s geografickými údaji o stanicích, jejichž polohu určujeme. Tyto údaje jsou dále v programu porovnávány s vypočtenou pozicí.

Zvláštní skupinu o jedné metodě tvoří *popisObvodOblastiDouble*. Název metody je opět zavádějící, protože byla původně určena k něčemu jinému a funkce se v průběhu programování změnila. Ve své konečné verzi přijme jako vstupní argument objekt typu *Area*, který vznikl průnikem oblastí výskytu, převede ho na polygon a spočítá

jeho těžiště pomocí vztahů[10]:

$$A = \frac{1}{2} \sum_{i=0}^{n-1} (x_i y_{i+1} - x_{i+1} y_i), \quad (5.4)$$

$$C_x = \frac{1}{6A} \sum_{i=0}^{n-1} (x_i + x_{i+1}) (x_i y_{i+1} - x_{i+1} y_i) \quad (5.5)$$

a

$$C_y = \frac{1}{6A} \sum_{i=0}^{n-1} (y_i + y_{i+1}) (x_i y_{i+1} - x_{i+1} y_i). \quad (5.6)$$

A značí obsah polygonu, n je počet vrcholů, x_i je souřadnice i -tého vrcholu na ose x a y_i je souřadnice téhož vrcholu na ose y . C_x a C_y značí souřadnici x a y těžiště polygonu.

5.3.2 Metoda *main* a průběh výpočtu cílových souřadnic

Jak je již napsáno výše, v metodě *main* je implementována funkcionalita metody CBG. Průchod touto metodou je zobrazen na vývojových diagramech D.1, 5.11 a D.2 (řazeno chronologicky). Na vývojových diagramech D.1 a D.2 jsou jen triviální záležitosti, jmenovitě příprava dat a výpis výsledků, proto jsou uvedeny v příloze D.

Po startu programu se vytvoří pole prvků *Cil*, které obsahuje všechny stanice, na něž bylo prováděno měření přenosového zpoždění. K tomuto účelu je použita metoda *NactiCile2*. Následně jsou pomocí metody *nastavZpozdeniCilumOstry* v cyklu všem prvkům *Cil* přiřazena pole obsahující příslušná naměřená přenosová zpoždění a komplementární objekty *Mapa*.

V další fázi se v cyklu pro všechny prvky pole *Cil* již provede odhad polohy stanice. To se skládá z několika dílčích kroků. Postupně se prochází pole s prvky *Mapa* a načítá se i příslušné naměřené zpoždění. To, pokud je menší než určitá hranice T_{tsh} , pak se pro výpočet vzdálenosti použije vzorec

$$d = RTT \cdot 100 \quad (5.7)$$

(geolokační metoda SOI). V opačném případě se počítá dle vzorce 2.1. Ve vztahu 5.7 je d vzdálenost a RTT naměřené obousměrné zpoždění. Proč je výsledná vzdálenost počítána zrovna takto bude vysvětleno v kap. 5.4.

V tuto chvíli je možné sestavit elipsu, jenž zastupuje oblast možného výskytu stanice. Do jejího konstruktoru

```
public Ellipse2D.Double(double x,double y,
                        double w, double h)
```

je za x dosazena zeměpisná délka, za y šířka. Parametr h značí výšku elipsy, použije se vypočtená vzdálenost z předchozího odstavce vynásobená korekční konstantou viz 4.3. Parametr w značí šířku elipsy a získá se pomocí vztahu

$$w = d \cdot \delta \cdot K,$$

kde d je vypočítaná vzdálenost, δ je korekce pro zkreslení na dané zeměpisné šířce a K je korekční konstanta, jejíž získání je popsáno v kap. 4.3.

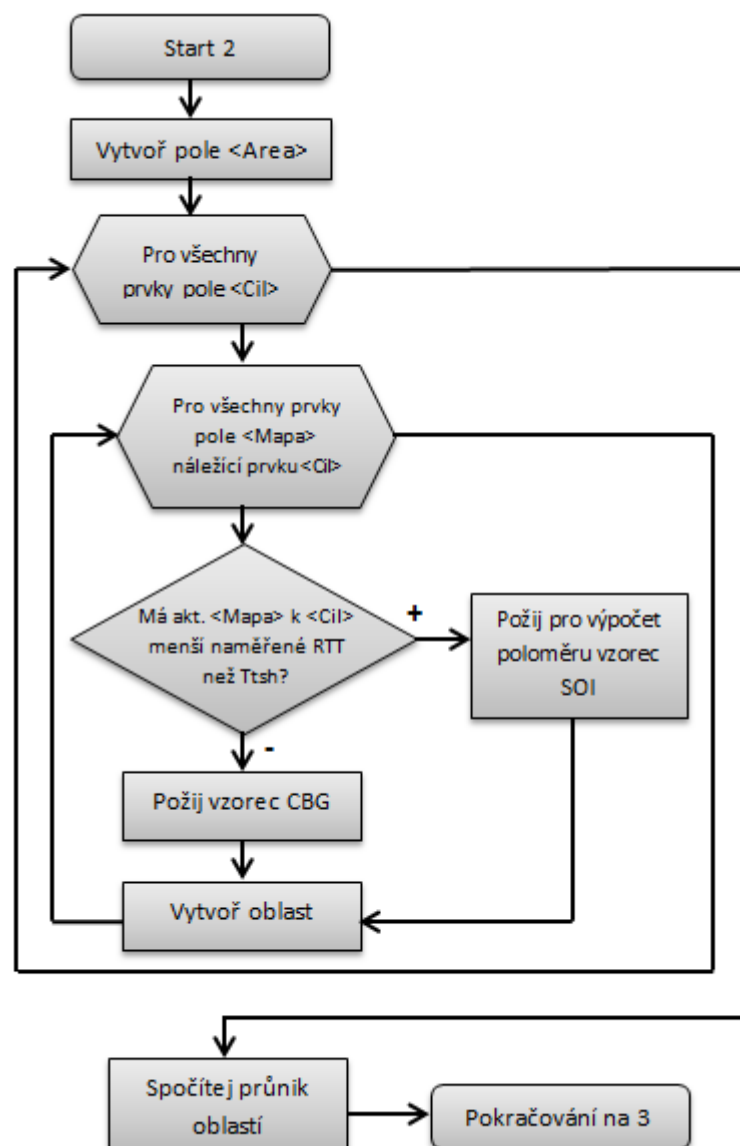
Všechny elipsy pro jeden *Cíl* jsou pak transformovány na oblast pomocí třídy *Area*.

```
public Area(Shape s).
```

Shape s zastupuje geometrický tvar, v tomto případě právě elipsu. Průnik je určen pomocí metody *intersect* náležící třídě *Area*.

Nad výsledným průnikem je zavolána metoda *popisObvodOblastiDouble*, která tuto oblast převede na polygon a vrátí souřadnice jeho těžiště. Výsledná chyba metody se spočítá pomocí rovnice Harvesine, viz 5.3.1, jako rozdíl známých souřadnic stanice a vypočítaných.

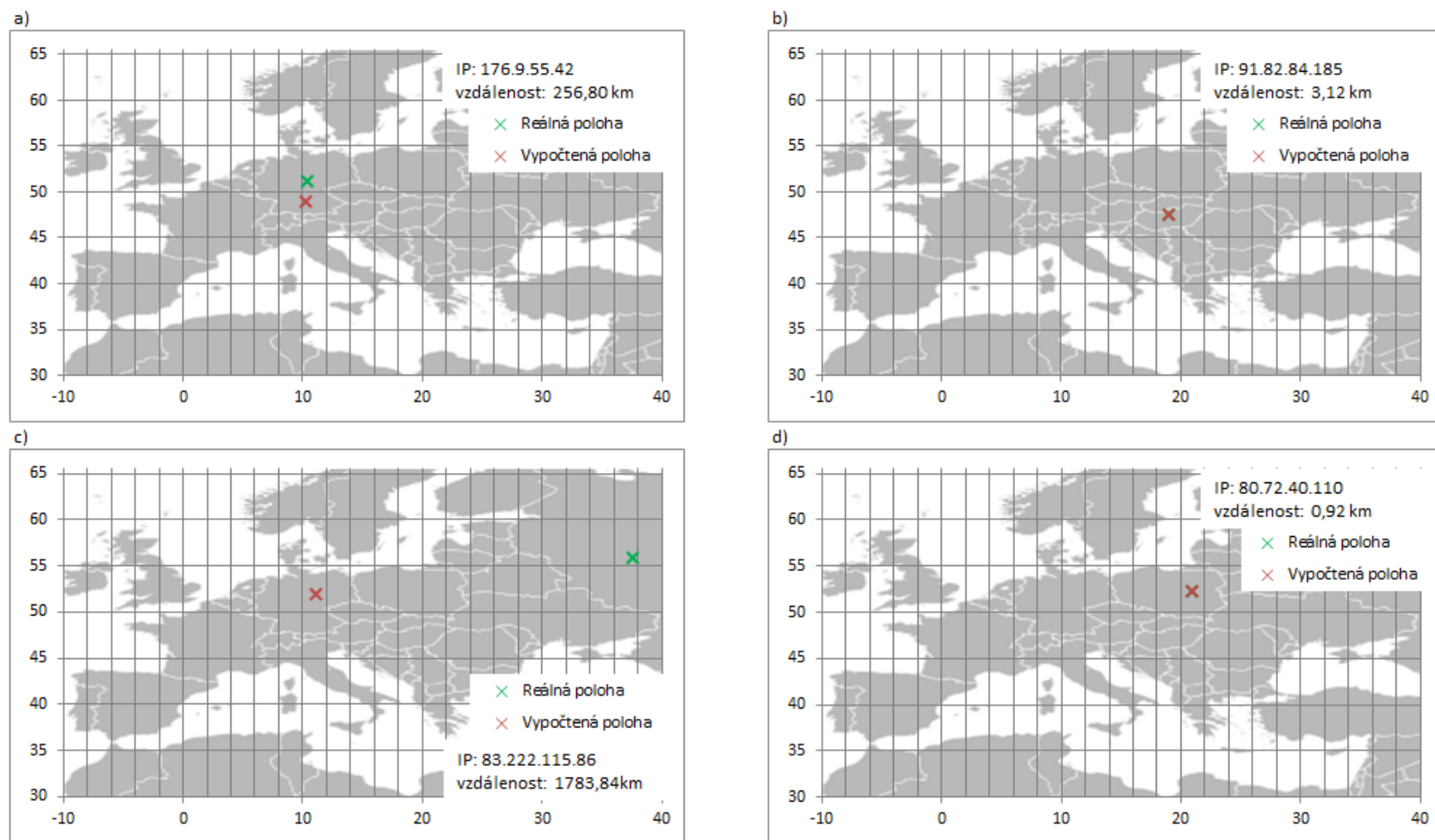
Dále následuje pouze formátování výsledku a jeho výpis do příkazové řádky. Výstup programu pro první čtyři stanice je vidět na obr. 5.12. Lokalizace těchto stanic je pak vidět na obr. 5.13, pro všechny ostatní stanice jsou pak obrázky na přiloženém médiu.



Obr. 5.11: Vývojový diagram střední části programu, práce s oblastmi

IP Adesa	Puvodni pozice		Vypoctena Pozice		Vzdalenost [km]
	E[°]	N[°]	E[°]	N[°]	
176.9.55.42	10,45153	51,16569	10,32859	48,85757	256,80181
91.82.84.185	19,04076	47,49841	19,0622	47,4744	3,1179
83.222.115.86	37,61842	55,75124	11,10705	51,81539	1783,64492
80.72.40.110	21,01223	52,22968	21,0069	52,2221	0,91728

Obr. 5.12: Záhloví a první čtyři řádky výstupu programu



Obr. 5.13: Lokalizace čtyř stanic

5.4 Porovnání výsledků metod SOI a CBG

Naměřená data byla použita pro určení polohy jak metodou CBG, tak metodou SOI⁴. Obě tyto metody navíc pracovaly se dvěma datasety. V prvním bylo provedeno měření z 24 uzlů, v druhém z 39. Tabulka E.1 obsahuje chybu s jakou byla určena poloha pro všechny kombinace metod a množství sond.

	SOI 24	CBG 24	SOI 39	CBG 39	SOI-CBG
	Chyba metody				
	km				
minimum	0,89	0,92	0,89	0,89	0,92
maximum	1955,38	5795,46	1702,41	4605,5	1783,84
průměr	312,93	537,46	269,98	505,70	244,21
medián	239,04	247,74	177,24	159,51	134,13

Tab. 5.1: Statistické údaje z chyb určených poloh

Z tab. 5.1 je patrné, a vyplývá to i z logiky věci, že se s přibývajícimi sondami zlepšuje přesnost (v tabulce klesají jak průměrná chyba a medián, tak i maximální odchylka) a že metoda CBG podává obecně lepší výsledky než metoda SOI. Vyšší průměrná chyba u metody CBG je zapříčiněna velkou chybou výpočtu u okrajového počtu stanic. Medián chyby, který má lepší vypovídající hodnotu je ze všech standardních geolokačních metod nejnižší pro CBG 39.

Kumulativní distribuční funkce pro všechny metody je na obr. 5.14. Stanice s chybou lokalizace větší, než 600 km byly prohlášeny za nelokalizovatelné (obr. 5.14 a)). Distribuční kumulativní funkce pro všechny stanice je na obr. 5.14 b).

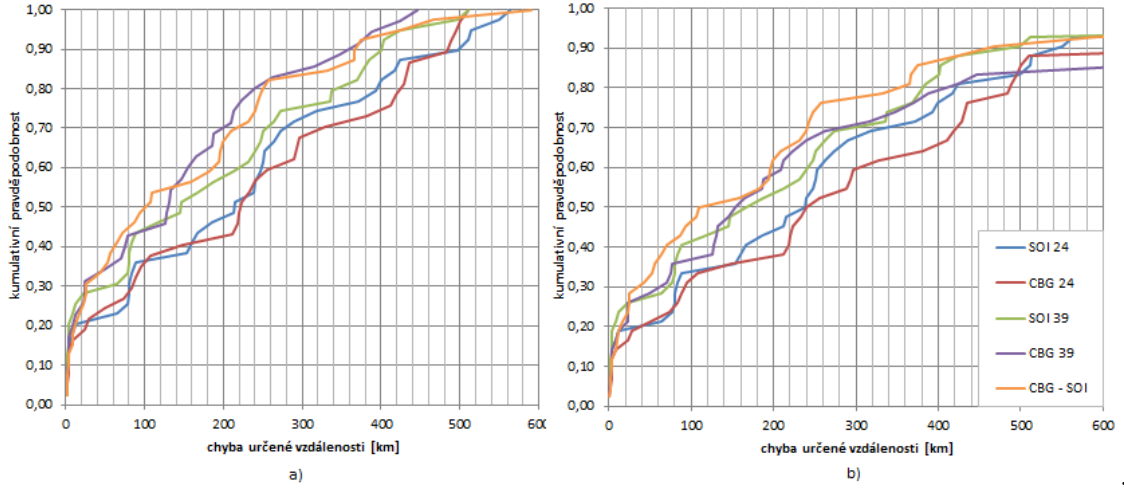
Z výstupních dat v tab. E.1 je ovšem vidět, že zlepšení není pro všechny stanice rovnoměrné ba naopak, chyba lokace některých stanic použitím „lepší“ CBG metody vzrostla. Důvody, které to mohly zapříčinit a jejich korekce jsou rozebrány v kap. 5.4.

5.4.1 Hybrid metod SOI a CBG

Je možné, že díky způsobu výsledného určení polohy došlo při výpočtu k podhodnocení vzdálenosti pro metodu CBG. Proto mohly některé výsledné chyby být horší než pro metodu SOI.

Z toho důvodu byl pokusně nastaven pro každý uzel (sondu) práh přenosového zpoždění. Pokud bylo libovolné naměřené zpoždění menší než tento práh, pak se

⁴Bylo k tomuto účelu použito vyvinutého programu, metoda SOI vyžadovala jen minimální zásahy do zdrojového kódu.



Obr. 5.14: Kumulativní distribuční funkce. a) S omezením vzdálenosti b) Bez omezení vzdálenosti

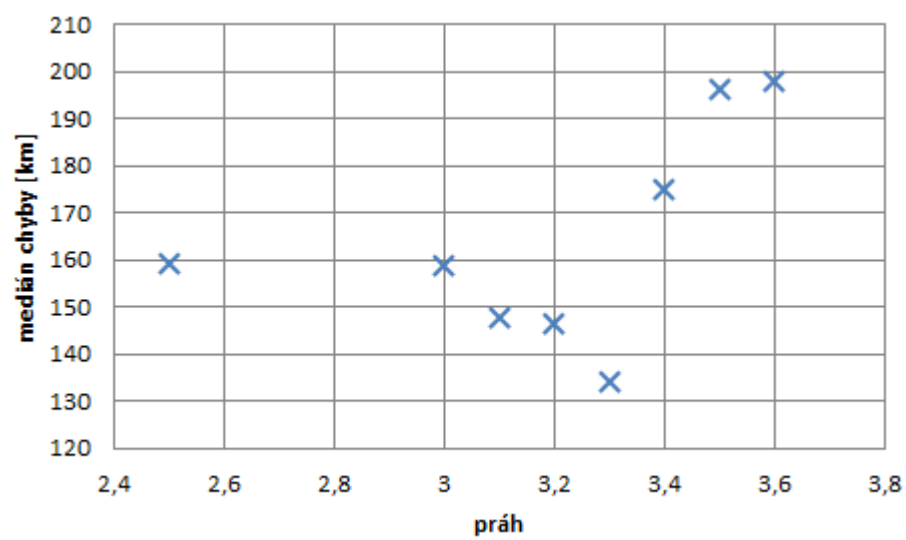
použila metoda SOI, v opačném případě CBG. Tato prahová hodnota T_{tsh} je získána jako

$$T_{tshi} = k \cdot a_i,$$

kde k je experimentálně získaná prahová konstanta a a_i je směrnice přímky (parametr $\langle a \rangle$) pro i -tý uzel (sondu, z níž bylo prováděno měření, v programu implementované pomocí třídy *Mapa*). V konečné implementaci bylo k zvoleno 3,3, důvody jsou posány níže.

Experimentální získání k probíhalo postupnou inkrementací jeho hodnoty od 1 („čistá“ CBG) po určitou maximální hodnotu. Maximální hodnota byla zvolena, tak že sledovaná veličina, jíž byl medián chyby určené polohy, dostatečně dlouho rostla (3 kroky za minimem). Krok byl volen nejdříve hrubý a v okolí minima byl zjemněn. Nejlepší výsledek podával program pro $k = 3,3$. Závislost mediánu chyby na k je vidět z grafu na obr. 5.15.

Konstanta k velmi pravděpodobně není univerzální a pro různé oblasti s různými množinami sond bude různá.

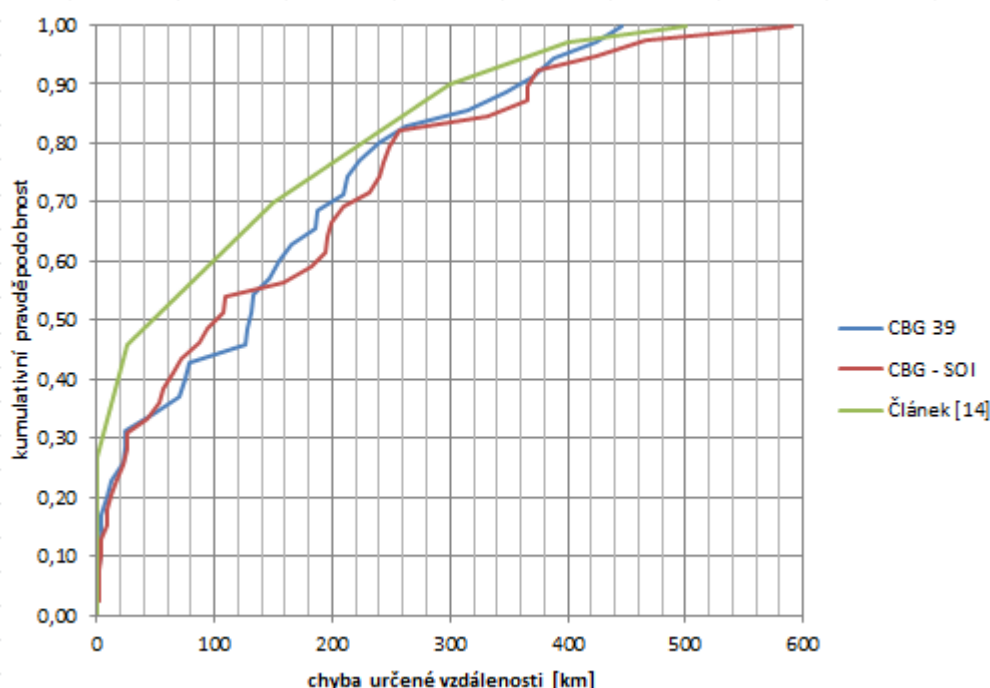


Obr. 5.15: Závislost mediánu chyby metody na prahové konstantě

6 DISKUSE VÝSLEDKŮ

Úkolem semestrálního projektu, který předcházel této práci, bylo implementovat geolokační metodu CBG a ověřit její funkci nad umělými daty. Funkcionalita programu, jenž je podrobně popsán v předchozí kapitole, je oproti verzi v semestrálním projektu upravená a rozšířená. V této kapitole jsou popsány výsledky, jichž bylo dosaženo s novou verzí programu a na datech jenž byla naměřena v dubnu 2013.

Ze všech implementovaných metod podávala nejlepší výsledek hybridní metoda CBG-SOI. Měla jak nejnižší průměr chyby, tak její medián.



Obr. 6.1: Porovnání implementovaných geolokačních metod s [14]

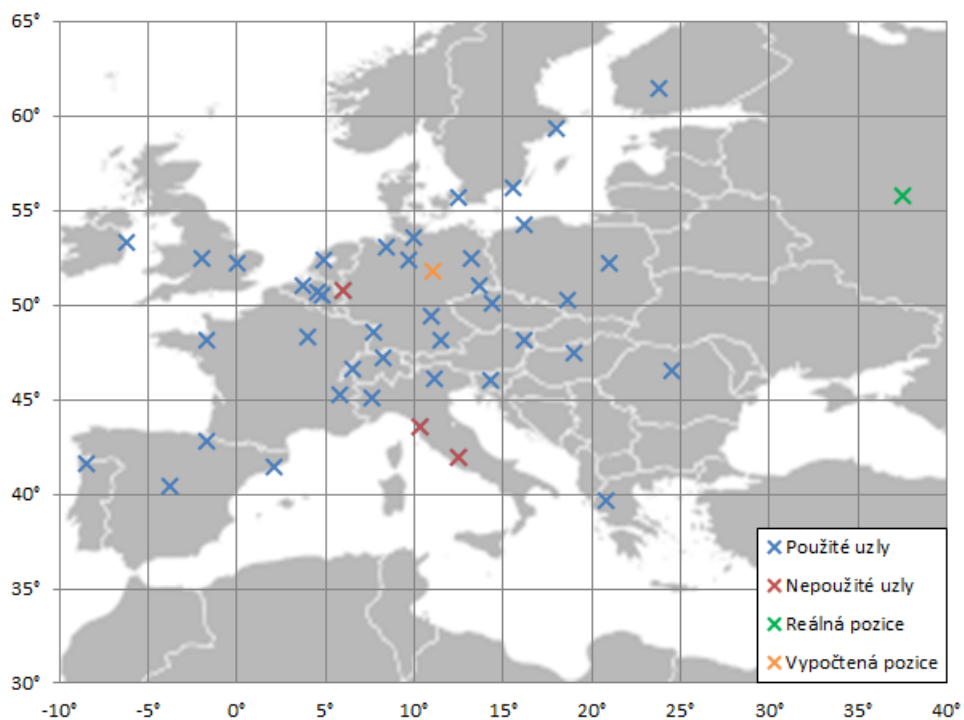
Na obr.6.1 jsou porovnány dvě metody, které podávaly v této práci nejlepší výsledky¹ s výstupem [14]. Jak je z onoho grafu patrné, obě porovnávané metody nedosahují takové přesnosti jako CBG implementovaná v [14].

Tento fakt může být zapříčiněn jak způsobem implementace (tato práce), tak vhodnou volbou stanic (tato práce i [14]). V práci [14] byla drtivá většina sond od 10° východní délky (v oné práci použili „42 landmarks in Western Europe“) na západ. V této práci byla pokryta i střední, jižní a část východní Evropy. To znamená pokrytí větší oblasti při srovnatelném množství sond.

Důležitý požadavek na dostatečně přesnou lokalizaci stanice je i její správná poloha vůči sondám. Na obr.6.2 je stanice s IP adresou 83.222.115.86 nacházející se

¹Výsledky implementovaných metod jsou zhodnoceny v kap. 5.4

na území Ruska. Tato stanice leží mimo síť sond a tento fakt se podepsal negativně na určení její pozice. Nejnižší chyba, se kterou byla určena je 1 702,4 km, nejvyšší pak 5 795,5 km. Výsledná pozice byla určena hluboko uvnitř sítě sond, blízko jejího středu. Podobně na tom jsou i další dvě stanice s IP adresami 193.136.163.66 a 77.47.133.22, i jejich poloha byla určena s chybou přesahující 1 000 km.



Obr. 6.2: Nevhodně zvolená stanice, která nemohla být díky poloze dostatečně přesně lokalizována

Tyto stanice byly prohlášeny za nelokalizovatelné, byť v tabulkách 5.1 a tabVysledkyMereni je s nimi počítáno při určení průměru, mediánu, minima a maxima. Po odebrání jich a všech ostatních stanic, u nichž chyba určení jejich pozice větší než 600 km z datasetu, dostaneme aktualizovanou tabulku 6.1.

Pro určení výsledné pozice stanice s dostatečnou chybou lze tedy formulovat několik požadavků:

- musí být měřitelné přenosové zpoždění od sond/y ke stanici,
- síť sond musí být pro danou oblast dostatečně hustá a rovnoměrná,
- stanice by měla být uvnitř sítě sond, ideálně co nejbliž jejímu prostředku.

První odrážka nepotřebuje komentáře. Co se týče hustoty sítě sond, ta závisí na oblasti a požadované chybě. Pro každou oblast je nutné ji zjistit experimentálně. Třetí odrážka vychází z dat naměřených a zpracovaných v této práci a důvody za jejím uvedením jsou popsány v této kapitole.

	SOI 24	CBG 24	SOI 39	CBG 39	SOI–CBG
	Chyba metody				
	km				
minimum	0,89	0,92	0,89	0,89	0,92
maximum	562,82	510,17	510,97	446,71	590,71
průměr	220,76	230,04	180,83	161,48	156,09
medián	214,96	223,52	146,93	130,50	106,43

Tab. 6.1: Statistické údaje z chyb určených poloh pro údaje s chybou menší než 600 km

7 ZÁVĚR

V práci byla rozebrána problematika zpoždění v síti, geolokačních metod a okrajově i mapových projekcí. Představena byla i platforma PlanetLab.

Implementované geolokační metody byly mezi sebou navzájem porovnány a obecně se potvrdilo, že metoda CBG podává lepší výsledky než metoda SOI. Po porovnání a diskusi výsledků byla vyvinuta metoda, která kombinuje vlastnosti obou předchozích. Princip funkce je jednoduchý, v závislosti na kalibračních datech pro každou sondu (uzel) a naměřeném přenosovém zpoždění se výsledná vzdálenost spočte pomocí jedné z nich. Poloměry oblastí, jejichž průnik je počítán jsou tedy získány buďto pomocí metody CBG nebo SOI.

Tato nová metoda nazvaná v této práci jako „hybrid SOI–CBG“ podávala nad naměřenými daty lepší výsledky než obě předchozí metody. Při porovnání s [14] podává horší výsledky, na rozdíl od této práce však byla metoda SOI–CBG používána na větším území (v podstatě celá Evropa oproti Západní) při srovnatelném množství sond.

Rozšířit práci by šlo do budoucna několika způsoby, pokusit se o zpřesnění výsledků, zlepšit automatizaci měření přenosového zpoždění a následného odhadu souřadnic.

Zlepšit výsledky by mohlo zhuštění nebo spíše zrovnoměnění sítě sond, která díky závislosti na síti Planetlab měla k ideálu daleko. Viz obr. 5.1, kde v Belgii jsou na relativně malém území tři sondy a na západě Francie nebo třeba na Balkánu je sond minimum. Dále by zlepšení přesnosti mohlo prospět použití přesnějšího převodu Země do 2D prostoru nebo použití geodetických knihoven.

Pod zlepšením automatizace si lze představit periodické měření parametrů sítě a využití těchto dat pro kalibraci. Dále by se program mohl dotázat na IP adresu a BASH skripty pomocí SSH a SCP na sondy nakopíroval, spustil a sám si naměřená data i zpracoval.

LITERATURA

- [1] Atan2. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001-2013, 1. 4. 2013 [cit. 2013-04-16]. Dostupné z: <<http://en.wikipedia.org/wiki/Atan2>>
- [2] BALEJ, Jiří a Dan KOMOSNÝ. Zdroje zpoždění při komunikaci v internetu. *Elektrorevue*. 2010, č. 42, s. 7. Dostupné z: <<http://www.elektrorevue.cz/cz/download/zdroje-zpozdeni-pri-komunikaci-v-internetu>>
- [3] BARAN, René. CentOS. *LinuxEXPRES: opravdový linuxový magazín* [online]. Brno: QCM, 2005 [cit. 2012-12-02]. Dostupné z: <<http://www.linuxexpres.cz/distribuce/centos>>
- [4] BOVY, C. J., et al. Analysis of End-to-end Delay Measurement in Internet [online]. In *Passive & Active Measurement: PAM 2002*, 2002. s. 8. Dostupné z URL:<<http://www-nas.et.tudelft.nl/people/Piet/papers/PAM2002.pdf>>.
- [5] Calculate distance, bearing and more between Latitude/Longitude points. *Movable Type Scripts* [online]. January 2010 [cit. 2013-04-16]. Dostupné z URL:<<http://www.movable-type.co.uk/scripts/latlong.html>>.
- [6] Comparison between TCP/IP and OSI. *Omnisecu* [online]. 2010 [cit. 2012-11-12]. Dostupné z URL: <<http://www.omnisecu.com/tcpip/tcpip-model.htm>>.
- [7] DAVIS, David. Layer 3 Switches Explained. *Happy router* [online]. 2007 [cit. 2012-11-13]. Dostupné z: <<http://happyrouter.com/layer-3-switches-explained>>.
- [8] Elipsa. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001-2013, 17. 9. 2012 [cit. 2012-12-08]. Dostupné z: <<http://cs.wikipedia.org/wiki/Elipsa>>.
- [9] ENDRUN TECHNOLOGIES. *Introduction to NTP*. Santa Rosa, Kalifornie, USA, 2004. Dostupné z: <<http://www.endruntechnologies.com/pdf/NTP-Intro.pdf>>.
- [10] Polygon. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001-2013, 25. 4. 2013 [cit. 2013-04-29]. Dostupné z: <<http://en.wikipedia.org/wiki/Polygon>>.

- [11] Equirectangular projection. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001-2012, 25.10.2012 [cit. 2012-12-08]. Dostupné z: <http://en.wikipedia.org/wiki/Equidistant_Cylindrical_Projection>.
- [12] FURUTI, Carlos A. *Map Projection Pages* [online]. 1996, 1997 [cit. 2012-12-02]. Dostupné z: <<http://www.progonos.com/furuti/MapProj/CartIndex/cartIndex.html>>.
- [13] FURUTI, Carlos A. A Simple Projection plus Two Derived Works: Deducing the Equidistant Cylindrical Projection. *MMap Projection Pages* [online]. 1996, 1997 [cit. 2012-12-03]. Dostupné z: <http://www.progonos.com/furuti/MapProj/Normal/CartHow/HowER_W12/howER_W12.html>.
- [14] GUEYE, Bamba, Artur ZIVIANY a Mark CROVELLA. Constraint-Based Geolocation of Internet Hosts. *IEEECAM TRANSACTIONS ON NETWORKING*. 2006, č. 14, s. 14.
- [15] HOLZHAUER, Florian. *IP Geolocation*. 2007, [cit. 2012-12-02], 11 s. Dostupné z: <http://faculty.ksu.edu.sa/ALFURAIH/Credit%20Card%20Fraud/IP_%20Geolocation.PDF>.
- [16] Map projection. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2012, 29-11-2012 [cit. 2012-12-02]. Dostupné z: <http://en.wikipedia.org/wiki/Map_projection#Cylindrical>.
- [17] PERCACCI, Roberto a Alessandro VESPIGNANI. Scale-free behavior of the Internet global performance. *Eur. Phys. J. B*. 2003, č. 32, s. 4. Dostupné z: <<http://arxiv.org/pdf/cond-mat/0209619.pdf>>
- [18] THE TRUSTEES OF PRINCETON UNIVERSITY. *Planetlab: An open platform for developin, deploying and accesing planetary-scale services* [online]. 2007, 2012-09-13 [cit. 2012-12-01]. Dostupné z: <<http://planet-lab.org/>>
- [19] THE TRUSTEES OF PRINCETON UNIVERSITY. *Planetlab Europe: An open platform for developin, deploying and accesing planetary-scale services* [online]. 2007, 2012-09-13 [cit. 2012-12-01]. Dostupné z: <<http://www.planet-lab.eu/>>

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

API Rozhraní pro programování aplikací – Application Programming Interface

CBG Constraint-Based Geolocation

EVP Ekvidistantní válcová projekce

GPS Globální družicový polohový systém – Global Positioning System

GUI Grafické uživatelské rozhraní – Graphic User Interface

HTTP Hypertext Transfer Protocol

LAN místní síť – Local Area Network

NTP Network Time Protocol

ICMP Internet Control Message Protocol

IP Internet Protocol

OS Operační systém

PLC PlanetLab Central

PLE PlanetLab Europe

RTT obousměrné zpoždění – Round Trip Time

SSH Secure Shell

SSID Service Set Identifier

TCP Transmission Control Protocol

TTL Množství prvků, kolika může IP paket projít, než je zahozen – Time to Live

UDP User Datagram Protocol

UML Unified Modeling Language

WAN Wide Area Network

SEZNAM PŘÍLOH

A	Obsah přiloženého média	60
B	Seznam uzlů sítě Planetlab	61
C	Seznam stanic pro účely geolokace	64
D	Vývojové diagramy	67
E	Chyba určené polohy pro geolokační metody	69

A OBSAH PŘILOŽENÉHO MÉDIA

- Elektronická verze práce
- Program v jazyce JAVA s daty použitými při ověřování funkce
- Seznam uzlů sítě Planetlab v souboru .xlsx
- Seznam stanic s vypočtenou polohou různými metodami v souboru .xlsx
- Soubor s vizuální lokalizací stanic

B SEZNAM UZLŮ SÍTĚ PLANETLAB

Institute	Zeměpisná délka [°]	Zeměpisná šířka [°]	host	IP
University of Vienna	16.24	48.16	ple01.fc.univie.ac.at	131.130.69.162
Warsaw University of Technology Europe	21.0069	52.2221	onelab4.warsaw.rd.tp.pl	195.116.53.24
Universite Catholique de Louvain	4.61667	50.6833	onelab3.info.ucl.ac.be	130.104.72.213
University of Cambridge	0.0917	52.2112	planetlab1.xeno.cl.cam.ac.uk	128.232.103.201
IRISA	-1.65071	48.1155	peeramide.irisa.fr	131.254.208.10
The Koszalin University of Technology	16.1972	54.2047	ple2.tu.koszalin.pl	62.108.171.76
Politecnico di Torino	7.670	45.07	planetlab3.di.unito.it	130.192.157.138
CC'TC / Universidade do Minho	-8.39722	41.5615	planetlab-um00.di.uminho.pt	193.136.19.13
Laboratoire d'Informatique de Grenoble	5.80872	45.2174	planetlab-1.imag.fr	129.88.70.226
Petru Maior University of Targu Mures	24.566	46.547	planetlab2.upm.ro	193.226.19.31
Universidad Publica de Navarra	-1.63544	42.7993	planetlab1.tlm.unavarra.es	130.206.158.138
IBBT	3.74	51.04	planck228ple.test.ibbt.be	193.191.148.228
Troyes University of Technology	4.06739	48.2693	planetlab2.utt.fr	194.254.215.12
Universitat Politècnica de Catalunya	2.11273	41.389	planetlab3.upc.es	147.83.30.164
Zuse Institute Berlin	13.2962	52.4545	planet2.zib.de	130.73.142.88
University of Ioannina	20.8386	39.6182	planetlab2.cs.uoi.gr	195.130.121.205
University of Ljubljana	14.3052	46.0319	planetlab1.fri.uni-lj.si	212.235.189.114
University College Dublin	-6.22169	53.3092	planetlab-node-01.ucd.ie	193.1.170.135
CESNET - Czech Education and Research Network	14.3916	50.102	planetlab2.cesnet.cz	195.113.161.83

KTH, Royal Institute of Technology, EE	18.067	59.3495	planetlab2.s3.kth.se	130.237.50.125
Blekinge Institute of Technology	15.5911	56.1819	itchy.comlab.bth.se	194.47.148.170
Eotvos Lorand University	19.0622	47.4744	planet1.elte.hu	157.181.175.249
Tampere University of Technology	23.855	61.449	planetlab1.rd.tut.fi	193.166.167.4
Quantavis	10.3252	43.5052	pierre.iet.unipi.it	131.114.53.187
Sapienza University of Rome	12.5037	41.891	planetlab-4.dis.uniroma1.it	151.100.59.9
FUNDP - The University of Namur	4.867	50.467	orval.infonet.fundp.ac.be	138.48.3.202
Technische Universitat Dresden	13.725	51.026	planet1.inf.tu-dresden.de	141.76.45.17
University of Neuchatel	6.56	46.59	planetlab1.unineuchatel.ch	192.42.43.22
Vrije Universiteit	4.9	52.35	planetlab1.cs.vu.nl	130.37.193.141
Friedrich-Alexander University Erlangen-Nuremberg	11.000	49.36	planetlab3.informatik.uni-erlangen.de	131.188.44.102
CREATE-NET International Research Center	11.155	46.067	plab2.create-net.org	193.206.22.134
University of Copenhagen, DIKU	12.5612	55.7014	planetlab1.diku.dk	192.38.109.143
Silesian University of Technology	18.6773	50.2901	plab2.ple.silweb.pl	83.230.127.122
University of Zurich, Communication Systems Group	8.32	47.23	planetlab1.csg.uzh.ch	192.41.135.218
HAW Hamburg	10.0386	53.5686	mars.planetlab.haw-hamburg.de	141.22.213.35
Universite de Strasbourg	7.73833	48.5237	planetlab1.u-strasbg.fr	130.79.48.55
RWTH Aachen	6.05	50.78	planetlab2.itsec.rwth-aachen.de	137.226.138.157
Aston University	-1.89098	52.4866	planetlab1.aston.ac.uk	134.151.255.180
Technische Universitaet Muenchen	11.5669	48.1493	planetlab2.lkn.ei.tum.de	138.246.99.250

L3S Research Center, University of Hannover	9.73	52.4	planet2.l3s.uni-hannover.de	83.246.92.212
Technical University of Madrid	-3.72	40.42	planetlab2.dit.upm.es	138.4.0.120
Jacobs University Bremen	8.49	53.04	planetlab2.eecs.jacobs-university.de	212.201.44.82

Tab. B.1: Seznam použitých uzlů sítě Planetlab

▽ na uzly označené tímto symbolem se nešlo připojit

C SEZNAM STANIC PRO ÚČELY GEOLOKACE

V této příloze je kompletní seznam uzlů, jejichž poloha měla být v rámci této práce ověřena. U několika uzlů nemohlo dojít ke změření jejich odevy a proto byly v tomto seznamu zachovány, jejich poloha nemohla být určena pomocí CBG. Tyto servery jsou v tab. C.1 označeny symbolem ∇ . Tento seznam byl vytvořen v týmu, jehož kromě mě byly součástí Henek Jan, Mrník Martin, Bc, Jelínek Ondřej a Pokorný Josef, Bc.

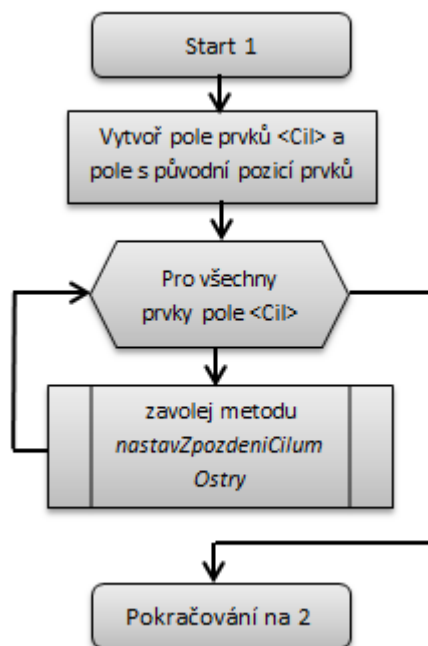
IP adresa	Poloha	Zeměpisná šířka [°]	Zeměpisná délka [°]
176.9.55.42	Germany	51.1656	10.4515
91.82.84.185	Budapest, Hungary, 05, Maďarsko	47.4984	19.0407
77.251.170.54 ∇	Heusden, 06, The Netherlands	51.7330	5.1382
83.222.115.86	Moscow, 48, Russia	55.7512	37.6184
80.72.40.110	Warsaw, 78, Poland	52.2296	21.0122
89.215.114.195 ∇	Plovdiv, 51, Bulgaria	42.1438	24.7495
109.70.149.49	Stroud, E6, UK	51.7457	-2.2177
85.11.157.54 ∇	Sofia, 42, Bulgaria	42.6964	23.3260
174.137.191.6	Amsterdam, The Netherlands	52.3702	4.8951
46.20.125.254	London, UK	51.5073	-0.1276
78.46.90.47	Gunzenhausen, Germany	49.1147	10.7541
88.198.19.202	Nuernberg, Germany	49.4520	11.0767
212.67.73.150	the Czech Republic, Prague 9	50.1074	14.5026
5.9.59.165	Germany, Frankfurt	50.1109	8.6821
213.153.32.170	Salzburg, Austria	47.8094	13.0550

217.79.128.22	Budapest, Hungary	47.4984	19.0407
194.146.252.199	Krakow, Poland	50.0646	19.9449
213.94.75.9	Vienna, Austria	48.2081	16.3738
94.136.136.2	Michalovce, Slovakia	48.7559	21.9148
89.216.2.122	Beograd, Serbia	44.8205	20.4622
91.90.160.3	Wroclaw, Poland	51.1078	17.0385
95.77.94.89	Bucharest, Romania	44.4377	26.0973
213.249.64.165	Amsterdam, The Netherlands	52.3702	4.8951
62.65.173.6	Bratislava, Slovakia	48.1462	17.1072
77.75.76.3	the Czech Republic, Prague 5	50.0715	14.4007
212.200.163.178	Kikinda, Serbia	45.8283	20.4652
85.236.100.91	London, UK	51.5073	-0.1276
188.165.34.68	Paris, France	48.8566	2.3522
88.190.22.159	Draveil, France	48.6853	2.4081
93.115.207.238	Jilava, Romania	44.3333	26.0833
81.201.56.141	the Czech Republic, Pilsen	49.7461	13.3642
193.166.3.2	Keilaranta 14, Espoo, Finland	60.2054	24.6558
130.251.19.2	Via Balbi 5, 16126 Genova, Italy	44.4151	8.9259
152.66.115.224	Muegyetem rkp. 9., H-1111 Budapest, Hungary.	47.4813	19.0563
132.252.181.87	Universitaetsstr. 1-15, 45117 Essen, Germany	51.4632	7.0042
129.240.8.200	Oslo, Norway	59.9399	10.7218

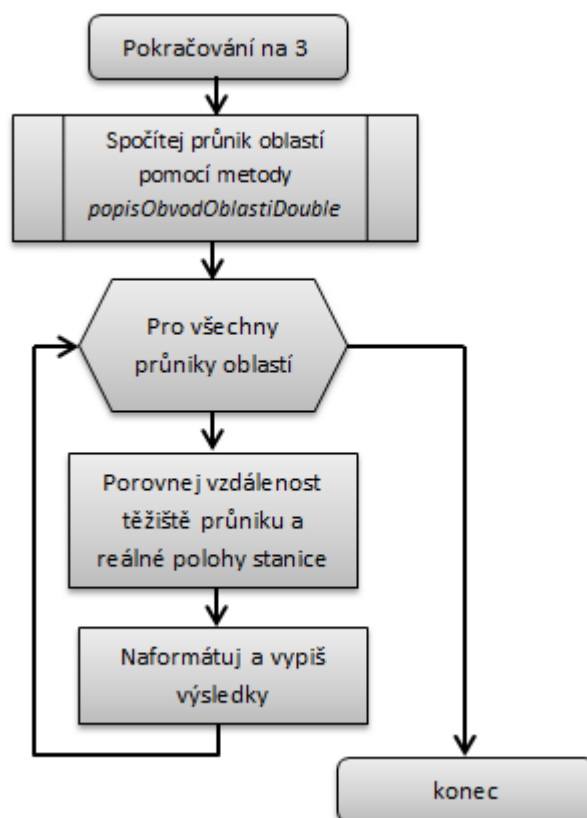
212.87.14.41	University of Warsaw, 00-927 Warsaw, Poland	52.2404	21.0192
130.239.141.10	Umea University, 901 87 Umea, Sweden	63.8205	20.3035
130.235.209.220	Lund University, S-222 40 Lund, Sweden	55.7108	13.20526
130.59.10.36	SWITCH, SWITCHmirror, CH-8021 Zurich, Switzerland	47.3686	8.53918
77.93.192.144▽	the Czech Republic, Brno - Židenice	49.2000	16.6333
81.2.194.154	the Czech Republic	49.7500	15.5000
91.235.52.167	Slovakia	48.6667	19.5000
131.175.187.11	Milano, Italy	45.4667	9.1939
129.27.201.245	Gratz, Germany	47.0693	15.4504
193.136.163.66	Lisabon, Portugal	38.7167	9.1333
109.70.148.245	London, UK	51.5142	0.0931
77.47.133.22	Kyiv, Ukraine	50.4333	30.5167

Tab. C.1: Seznam stanic pro účely geolokace

D VÝVOJOVÉ DIAGRAMY



Obr. D.1: Vývojový diagram úvodní části programu, příprava polí



Obr. D.2: Vývojový diagram koncové části programu, počítání chyby metody a výpis výsledků

E CHYBA URČENÉ POLOHY PRO GEOLO- KAČNÍ METODY

IP adresa	SOI 24	CBG 24	SOI 39	CBG 39	SOI-CBG
	Chyba metody				
	[km]				
176.9.55.42	562,82	484,11	400,73	4605,56	256,80
91.82.84.185	3,12	3,12	3,12	3,12	3,12
83.222.115.86	1955,38	5795,46	1702,41	1722,16	1783,84
80.72.40.110	0,92	0,92	0,92	0,92	0,92
109.70.149.49	166,42	435,52	166,42	164,86	247,81
174.137.191.6	237,87	220,28	2,27	130,50	25,11
46.20.125.254	79,70	223,52	79,70	70,58	198,76
78.46.90.47	392,82	852,98	384,98	4261,96	42,09
88.198.19.202	371,00	28,47	230,77	132,19	93,95
212.67.73.150	7,94	7,94	7,94	7,94	7,94
5.9.59.165	185,44	292,60	368,65	208,53	182,67
213.153.32.170	240,21	240,21	240,21	240,21	240,21
217.79.128.22	3,12	3,12	3,12	3,12	3,12
194.146.252.199	251,07	148,23	251,07	2489,49	230,86
213.94.75.9	11,27	50,24	11,27	11,27	11,27
94.136.136.2	423,58	411,10	423,58	423,58	423,58
89.216.2.122	399,57	510,17	402,68	446,71	466,84
91.90.160.3	153,71	108,08	146,93	154,16	195,39
95.77.94.89	263,16	4197,98	263,16	315,60	366,09
213.249.64.165	252,07	327,83	2,27	23,68	16,30
62.65.173.6	64,35	73,84	64,35	74,86	64,35
77.75.76.3	3,45	3,45	3,45	3,45	9,37
212.200.163.178	212,05	211,29	212,05	212,05	193,47
85.236.100.91	79,70	95,38	79,70	127,43	55,60
188.165.34.68	289,46	288,59	335,41	222,75	52,54
88.190.22.159	214,96	381,01	117,17	49,08	209,44
93.115.207.238	273,12	296,20	273,12	387,35	331,86
81.201.56.141	83,52	83,52	83,52	125,22	71,11
193.166.3.2	159,26	255,27	144,92	185,58	158,60
130.251.19.2	513,31	431,74	375,61	187,07	86,20
152.66.115.224	0,89	23,36	0,89	0,89	1,72

132.252.181.87	549,98	495,24	188,07	145,93	106,43
129.240.8.200	417,71	428,75	337,81	370,59	365,32
212.87.14.41	2,20	2,20	2,20	2,20	2,20
130.239.141.10	510,97	487,90	510,97	648,75	590,71
130.235.209.220	497,65	500,57	497,65	347,45	375,00
130.59.10.36	318,13	417,99	22,60	22,60	22,60
81.2.194.154	88,47	88,47	88,47	23,71	25,36
91.235.52.167	247,09	233,17	247,09	261,70	243,20
193.136.163.66	1520,57	1387,78	1520,57	1238,16	1257,19
109.70.148.245	77,50	218,10	77,50	77,65	109,66
77.47.133.22	1057,36	1827,73	1063,86	1108,81	1128,04
minimum	0,89	0,92	0,89	0,89	0,92
maximum	1955,38	5795,46	1702,41	4605,5	1783,84
průměr	312,93	537,46	269,98	505,70	244,21
medián	239,04	247,74	177,24	159,51	134,13

Tab. E.1: Chyba určení výsledných souřadnic